PCT WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 7 : G07F 7/10, G06F 1/00, 12/14

A1

(11) International Publication Number: WO 00/26866

(43) International Publication Date: 11 May 2000 (11.05.00)

(21) International Application Number: PCT/CA99/01011

(22) International Filing Date: 29 October 1999 (29.10.99)

(30) Priority Data:
2,250,499 30 October 1998 (30.10.98) CA

(71) Applicant *(for all designated States except US)*: MOTUS TECHNOLOGIES INC. [CA/CA]; 390, rue Saint-Vallier Est, Bureau 100, Québec, Québec G1K 3P6 (CA).

(72) Inventor; and
(75) Inventor/Applicant *(for US only)*: DURANT, Pierre [CA/CA]; 173, rue Pierre-Constantin, Saint-Augustin-de-Desmaures, Québec G3A 2V3 (CA).

(74) Agent: ROBIC; 55 St. Jacques, Montréal, Québec H2Y 3X2 (CA).

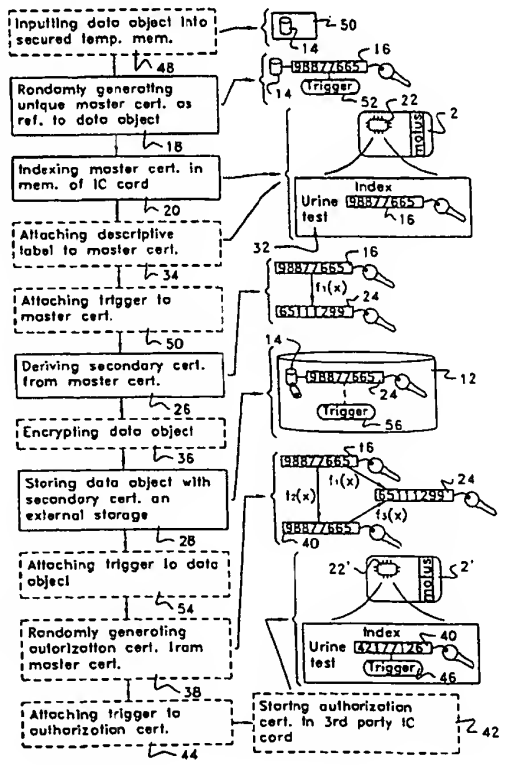
(81) Designated States: AE, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CU, CZ, DE, DK, DM, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, UA, UG, US, UZ, VN, YU, ZA, ZW, ARIPO patent (GH, GM, KE, LS, MW, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

Published
With international search report.
Before the expiration of the time limit for amending the claims and to be republished in the event of the receipt of amendments.

(54) Title: SECURE MEMORY EXPANSION OF AN IC PORTABLE DEVICE

(57) Abstract

The method is for securely expanding a storage capacity of an IC portable device's memory. For a given data object stored or to be stored in the memory of the IC portable device, a unique master certificate, used as a descriptor of the data object, is generated randomly. The master certificate is indexed in the memory of the portable device. A secondary certificate is derived from and is determinable only using the master certificate. The data object is stored with the secondary certificate on a data storage medium external to the portable device. The memory of the smart card is thus freed from the data object and only the IC portable device has key information to retrieve the data object stored on the data storage medium.



FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AL	Albania	ES	Spain	LS	Lesotho	SI	Slovenia
AM	Armenia	FI	Finland	LT	Lithuania	SK	Slovakia
AT	Austria	FR	France	LU	Luxembourg	SN	Senegal
AU	Australia	GA	Gabon	LV	Latvia	SZ	Swaziland
AZ	Azerbaijan	GB	United Kingdom	MC	Monaco	TD	Chad
BA	Bosnia and Herzegovina	GE	Georgia	MD	Republic of Moldova	TG	Togo
BB	Barbados	GH	Ghana	MG	Madagascar	TJ	Tajikistan
BE	Belgium	GN	Guinea	MK	The former Yugoslav Republic of Macedonia	TM	Turkmenistan
BF	Burkina Faso	GR	Greece			TR	Turkey
BG	Bulgaria	HU	Hungary	ML	Mali	TT	Trinidad and Tobago
BJ	Benin	IE	Ireland	MN	Mongolia	UA	Ukraine
BR	Brazil	IL	Israel	MR	Mauritania	UG	Uganda
BY	Belarus	IS	Iceland	MW	Malawi	US	United States of America
CA	Canada	IT	Italy	MX	Mexico	UZ	Uzbekistan
CF	Central African Republic	JP	Japan	NE	Niger	VN	Viet Nam
CG	Congo	KE	Kenya	NL	Netherlands	YU	Yugoslavia
CH	Switzerland	KG	Kyrgyzstan	NO	Norway	ZW	Zimbabwe
CI	Côte d'Ivoire	KP	Democratic People's Republic of Korea	NZ	New Zealand		
CM	Cameroon			PL	Poland		
CN	China	KR	Republic of Korea	PT	Portugal		
CU	Cuba	KZ	Kazakstan	RO	Romania		
CZ	Czech Republic	LC	Saint Lucia	RU	Russian Federation		
DE	Germany	LI	Liechtenstein	SD	Sudan		
DK	Denmark	LK	Sri Lanka	SE	Sweden		
EE	Estonia	LR	Liberia	SG	Singapore		

SECURE MEMORY EXPANSION OF AN IC PORTABLE DEVICE

FIELD OF THE INVENTION

The present invention relates to memory management processes involving IC portable devices, and more particularly to a method of securely expanding a storage capacity of a memory of an IC portable device, e.g. a smart card.

BACKGROUND

Smart cards and other IC portable devices have limited memory space/capacity (e.g. 1-8 kb), often restricted by the physical dimensions of the devices and the technology. Compression methods have been used to fit more data in a given memory space. However, the memory capacity is nevertheless reached sooner or later. Other types of memory storage mediums added on a smart card have also been proposed, like optical storage medium, etc. These storage mediums require additional circuitry for processing the data stored on them, notwithstanding the restrictions, limitations or incompatibility caused by the technology.

Because of the memory limitation of smart cards, they are rather generally used as mere personal identification devices to access data on an external mass storage medium, with access rights depending on the status of the smart card holder. For example, URLs or medical examination numbers may be stored in a smart card, thereby providing references to externally stored data on a web site, ordinary databank, the references and the data being managed by the external software applications and not the smart card. As smart cards use encryption to secure the data, they are increasingly used for financial transactions, where the information to be

stored in the cards is limited to few numbers and words, if any.

Still, information storage is limited to the memory space afforded by the smart card. Thus, the memory limitation of smart cards is an obstacle to their use in many fields of application where they would be of great use. An example of such fields of application is the medical field where the large scale implementation of private portable files representing the medical history and profile of the patient is presently hardly conceivable due to numerous difficulties despite the efforts of many to devise solutions.

A key factor with smart cards is and remains the data security.

SUMMARY

An object of the invention is to provide a method of securely expanding a storage capacity of a memory of an IC portable device.

A subsidiary object of the invention is to solve the memory limit problem with IC portable devices while maintaining data privacy even on remote storage site.

A subsidiary object of the invention is to provide a method in which data stored remotely from the IC portable device are linked thereto by indirect, preferably unreversably derived relationship, and in which the references to the data never leave a secure, delimited memory zone area distinct from the external medium in which the data are remotely stored.

A subsidiary object of the invention is to provide a method by which access authorization to selected remotely stored data can be given to a third party without ever endangering the original link privacy of the data with the proprietary IC portable device.

A subsidiary object of the invention is to provide a method by which the data stored externally to the IC portable device can be deprived of any character without the IC portable device, and can be anonymous and depersonalized.

A subsidiary object of the invention is to provide a method by which an IC portable device can generate secure virtual memory for its own needs.

A subsidiary object of the invention is to provide a method that preserves the advanced security features provided by an IC portable device.

A subsidiary object of the invention is to provide a method by which remotely stored data objects are managed solely under the control of an IC portable device.

According to the invention, there is provided a method of securely expanding a storage capacity of a memory of an IC portable device, comprising the steps of, for a data object stored or to be stored in the memory of the IC portable device:

randomly generating a unique master certificate used as a reference to the data object;

indexing the master certificate in the memory of the IC portable device;

deriving a secondary certificate from the master certificate, the secondary certificate being determinable only using the master certificate;

storing the data object with the secondary certificate on a data storage medium external to the IC portable device;

whereby the memory of the smart card is freed from the data object as the data object is stored on the data storage medium, thereby securely expanding the storage capacity of the memory of the IC portable device as only the IC portable device has key information to retrieve the data object stored on the data storage medium.

According to an aspect of the invention, there is provided a method of operating an IC portable device having a memory for storing data objects, comprising the steps of:

detecting a predetermined condition relative to the memory of the IC portable device; and

if the condition is detected, applying the aforesaid memory expanding method on a number of the data objects stored in the memory of the IC portable device.

The method according to the invention relies upon IC portable device technology to store and manage anonymous references to anonymous data located on remote sites (data warehouses). The method allows for the secure storage and use (access control) of the information. Only the IC portable device carrying the master certificates is capable to locate, identify and restore the information stored in an anonymous fashion and possibly encrypted in the remote sites. The method allows for the secure management of any type of information that requires a secure storage and access control. The method also allows the device to generate virtual memory to meet its needs.

BRIEF DESCRIPTION OF THE DRAWINGS

A detailed description of preferred embodiments will be given herein below with reference to the following drawings, in which like numbers refer to like elements:

Figure 1 is a schematic diagram showing a system suitable for the working of the invention;

Figure 2 is a flow chart illustrating the method according to the invention;

Figure 3 is a schematic diagram showing a rudimentary architecture of a system according to the invention;

Figure 4 is a schematic diagram showing the conceptual organization of a card configured according to the invention;

Figure 5 is a flow chart illustrating a representative process for reading and writing operations in a system according to the invention;

Figure 6 is a flow chart illustrating details of a representative writing process according to the invention;

Figure 7 is a flow chart illustrating details of a writing process on a remote database according to the invention;

Figure 8 is a flow chart illustrating details of a data screening process according to the invention;

Figure 9 is a flow chart illustrating details of a representative reading process according to the invention;

Figure 10 is a flow chart illustrating details of a reading process on a remote database according to the invention;

Figure 11 is a flow chart illustrating a search process for a data object stored on a remote database according to the invention;

Figure 12 is a schematic diagram depicting an initial state of a system according to the invention;

Figure 13 is a schematic diagram depicting a writing of a data object in the card according to the invention;

Figure 14 is a schematic diagram depicting a writing of a data object in a remote database according to the invention;

Figure 15 is a schematic diagram depicting a writing of a data object when a card has insufficient memory space according to the invention;

Figure 16 is a schematic diagram depicting a screening of data objects in the card according to the invention;

Figure 17 is a schematic diagram depicting a purge of data objects in the card according to the invention;

Figure 18 is a schematic diagram depicting a reading of a data object in a card according to the invention;

Figure 19 is a schematic diagram depicting a reading of an index in a card according to the invention;

Figure 20 is a schematic diagram depicting a reading of an indexed data object according to the invention; and

Figure 21 is a schematic diagram depicting a reading of a data object on a remote database as a result of a forced transfer, according to the invention.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

Referring to Figure 1, there is shown a schematic diagram of a system suitable for the working of the invention, which resides in securely expanding a storage capacity of a memory of an IC portable device. In the illustrated embodiment and the following description, the IC portable device is a smart card 2. However, it should be understood that other types of IC portable devices can be used, e.g. a tag, without departing from the invention. The system includes a computer 4 provided with a reader 6 for communicating with the smart card 2. The computer 4 can be considered as being on a client side of the system as it generates and sends requests to a server 8 according to a given protocol, asking for information or action, and the server 8 responds thereto. There may be either one centralised server or several distributed ones, as illustrated, interconnected together through a network 10, e.g. internet. The reader 6 can be part of a SAM (Secure Access Module) providing many functions and features especially built for communication and operation with a smart card. The SAM may be conveniently designed to perform a multitasking management of transactions between an application software in the computer 4 and the card 2, compression/decompression of data, logical access management of the physical memory zones on the card 2, in addition to an

adaptation to different models of cards via a secured link that prevents data misappropriation.

The usual purpose of the computer 4 / reader 6 / smart card 2 arrangement is to manage data stored directly in the card 2 or in the computer's storage device (e.g. hard disk) relative in such a case to reference data stored in the card 2. Additional resources can be obtained by opening the arrangement to the network 10 and servers 8 providing remote data storage media 12 to store some or all the data.

However, by operating such a system using conventional methods and techniques, the storage capacity of the memory 22 (see figure 2) of the smart card 2 is nevertheless reached sooner or later, or the data stored in the remote storage media 12 or the communications between the reader 6 and the servers 8 are unsecured, or yet the system is subjected to other deficiencies. The present invention provides a method that solves these and other problems, as it will become apparent hereinafter.

In brief, the method resides in storing data objects that would otherwise be normally stored in the card 2, on a remote storage medium and indexing in the card 2 in a special secure manner, the information needed to trace and retrieve the data objects. In view of these functions, the card 2 can so to speak be referred to as an index card while the method can be referred to as an indexing method. The method also relies upon special processes to manage the storing of data objects in the card 2 or on a remote database 12, to transfer data objects from the card 2 to the remote database 12 depending on specific conditions, and to make the data objects stored on the remote server 8 unusable without the card 2. These features are all described in detail hereinafter.

Referring to Figure 2, there is shown a flow chart illustrating the method according to the invention. The

method is applicable either to a data object 14 already stored or to be stored in the memory 22 of the IC card 2. In the second case, the data object 14 can be stored in the memory 22 of the IC card 2 before applying the method, or the method can simply be applied directly on the data object 14 to be processed (so it is stored directly in a remote database), which can be advantageous when the data object 14 is large (it saves time). As depicted by block 18, the method has a step of randomly generating a unique master certificate 16 used as a reference to the data object 14. The certificate involved in this step is somewhat analogous to certificates found in cryptographic systems in that it can be generated using a secret key and other proprietary information in the card 2, when it is generated by the card 2. As depicted by block 20, the master certificate 16 is indexed in the memory 22 of the IC portable device 2. As depicted by block 26, a secondary certificate 24 is derived from the master certificate 16, the secondary certificate 24 being determinable only using the master certificate 16, or an intermediary certificate derived from the master certificate 16. Thus, the function $f_1(x)$ used to derive the secondary certificate 24 should preferably be a one-way function. As depicted by block 28, the data object 14 is stored with the secondary certificate 24 on a data storage medium 30 external to the IC portable device 2. The secondary certificate 24 acts so to speak as a tag to localize the data object provided that the master certificate 16 is used in doing so. The data storage medium 30 can be provided by one of the servers 8 (shown in figure 1) where the data object 14 is saved in the database 74, 76, 78. The memory 22 of the smart card 2 is thus freed from the data object 14 as the data object 14 is stored on the data storage medium 12, thereby securely expanding the storage capacity of the memory 22 of

the card 2 as only the card 2 has key information (i.e. the master certificate 16 to compute the secondary certificate 24) to retrieve the data object 14 stored on the data storage medium 12.

As depicted by block 34, a descriptive label 32 can be attached to the master certificate 16, the descriptive label 32 representing a character of the data object 14. In a healthcare application, the descriptive label 32 may relate for example to a health care type of service. The purpose of such a descriptive label 32 is to enable the retrieval of a specific type of data objects 14 stored in the data storage medium 30 without requiring the retrieval of all of the data objects 14 and their analysis to determine whether they relate to the desired specific type or not. The descriptive label 32 is indexed with the master certificate 16 in the memory 22 of the card 2.

As depicted by block 36, the data object 14 can be encrypted prior to the storing step (block 28), the data object 14 stored on the data storage medium being thus encrypted. The encryption of the data object 14 is particularly desirable if the database used is unsecured, as it is the case for common databases. However, other security measures can be embodied instead of encryption, some of which will be described hereinafter.

In medical and other applications, it might be desirable to provide a way to authorize a third party, e.g. a physician, to have access to certain of the data objects, e.g. results of tests carried out at a remote site, for a given patient who is the holder or owner of the smart card 2. In such instance, as depicted by block 38, there is a step of randomly generating a data access authorization certificate 40 derived from the master certificate 16 and with which the secondary certificate 24 is determinable. As depicted by block 42, the authorization certificate 40 is then stored in

the IC portable device held by the third party. The data access authorization certificate 40 can be generated by an appropriate one-way function $f_2(x)$ applied on the master certificate 16. The secondary certificate 24 can then be determined by applying another one-way function $f_3(x)$ using the authorization certificate 40. The data object 14 stored on the data storage medium 30 is thereby retrievable with the card of the third party.

The authorization certificate 40 is preferably transmitted to the third party's card via a secured communication channel with the card 2 from which the master certificate 16 originates, using a cryptographic protocol.

As depicted by block 44, a trigger 46 can be attached to the authorization certificate 40, the trigger 46 carrying authorization use instructions regarding the access to the data objects by the third party. For example, the instructions may be selected among a life duration of the authorization certificate 40 (the certificate can be automatically destroyed after a predetermined time period), a service location restriction (to restrict the stations where the third party can have access to the data), a third party class restriction (to prevent access to the data for selected types of third parties), and a transitive authorization control (to enable the third party to authorize another IC card holder to have access to the data). The concept of triggers can be implemented for other tasks, like the management of data objects in the card 2.

A secured microprocessor card server that can be embodied in the card reader 6 preferably manages the certificates 24, 40.

The master and secondary certificates 16, 24, and the other certificates like the authorization certificate 40, can be generated using a cryptographic algorithm like DES (Data

Encryption Standard). Thus the functions $f_1(x)$, $f_2(x)$, $f_3(x)$ can be based on the DES algorithm. Other ways to generate the certificates 16, 24, 40 can be used as well. However, the certificates 16, 24, 40 must possess certain properties. They must be unique for a same data storage medium 12. They must be random, so that there is no link between them. They must be anonymous, so that they do not provide any known link with the card owner. Their binary sequence must be long enough to provide an important set of references to avoid attacks based on the systematic generation of all the binary sequences. The certificate generation can be implemented locally in each of the cards, or in a centralized fashion relative to the server 8 providing the data storage medium 12.

If locally implemented in the card 2, the certificate generation process can be integrated to the card's OS (which may be virtual in the case where a SAM is used, or even distributed among secured resources). A certificate thus originates from the card 2. The difficulty with this approach resides in the coordination of the generation of the certificates between the different cards of the system to avoid collisions. For this purpose, a generation primary root called primary master certificate can be integrated in each card. The root is so to speak analogous to a unique secret key. From this point of view, the card 2 uses this root in the embodiment of a cryptographic derivation device. This certificate is qualified as primary master, since all the certificates of the card are descendants therefrom. The direct descendants can be qualified as secondary master certificates. These are the certificates 24 that are generated during a transaction and associated to a data object. The secondary master certificates are behind the generation of external derived certificates (transient certificates that can be used to label temporarily the

information in transit or permanent derived certificates like the secondary certificates 24 used in the data warehouses like those provided by the servers 8). The primary and secondary master certificates preferably never leave the card (in its memory or a secured memory zone distinct from the data storage medium 12) for security purposes. The certificate derivation can be handled in many ways, involving additional parameters that can be used to strengthen the security of the data eventually stored in the data warehouse 12, and prevent reverse personalization of the data by usual processes.

In the case of a centralized management of the certificates, each data warehouse 12 is preferably provided with a certificate generator 84 that coordinates the certificate distribution in its reference domain. With this approach, there is no certificate lineage notion with respect to the primary root of the card 2. Indeed, it is not necessary to implement a certificate derivation process that coordinates the generation sequence relative to an entire index card set to avoid collisions of sequences. However, certificate derivation can nevertheless be used to preserve the anonymity of the certificates obtained by a card. In this sense, there is still a set of certificates that qualify as master certificates, behind the external certificates. Unlike the secondary master certificates of the former approach, which are linked by their ascendants, the primary master certificate and the master certificates in the later approach share no relationships and are absolutely random. This feature improves the second criterion associated to the certificates (the randomness), and reinforces the global security of the data. Indeed, a central certificate generator 84 allocates certificates in a random fashion as a function of the requests made by the index cards. The allocation of certificates by a generator can be achieved according to

various strategies, e.g. sequential allocation, random selection of certificates in a predetermined set, collision with trial and error, etc. In a system with several distributed data storage media 12 associated with servers 8, each medium 12 can be identified from the others by a unique reference domain identifier. The master certificates can then be randomly generated, using the unique reference domain identifier, by the central certificate generator 84, and transmitted to the card 2 via a secured link. Other master certificate generation processes can be used,

The process of deriving a secondary certificate from the primary certificate, as depicted by the block 26, may take various forms, depending on the implemented strategy. The term "deriving" in this context is not limited to its mathematical meaning. It can be any data processing that generates an information which originates from the master certificate, whose form is also not limited to numerals (it can be a word, a number, a phrase, etc.). The secondary certificate can be the result of multiple derivation steps combined or not with other functions (e.g. hash functions). As an example, let C0 be the master certificate. Using a predetermined (mathematical) algorithm combined with C0, there is obtained a first certificate C1 derived from C0. Using the algorithm combined with C1, there is obtained a second certificate C2 derived from C1, and so on. If the algorithm is a one-way function, then any one of C1 to CN can be used to provide the secondary certificate. Otherwise, at a given stage, a one-way function should be applied to break the sequence and prevent reverse derivation that would yield back C0 from a given CN. By using such a strategy, then an intermediate Ci could be stored in a third party's card to authorize data access without ever being capable to trace back the original certificate C0 stored in the user's card, thereby maintaining the system's security. The original

derivation sequence can be branched to produce different intermediate certificates that can be stored in different third parties' cards without endangering the data access privacy of any one of the third parties as no one could step back to then follow the other branch of the sequence of another third party.

As depicted by block 48, the data object 14 can be inputted into a secured temporary memory 50 provided in the card reader 6 or the computer 4, in response to a software application request, prior to the generation of the master certificate 16 (block 18). This then starts the secure processing of the data object 14 throughout the steps of the indexing method.

Referring to figure 1, the card 2 may comprise a database of actions 30 in relation with conditions associated thereto, and triggers that trigger execution of each action for which the associated condition is met. The database of actions 30 can be provided in the computer 4, which acts as a card server. Triggers are static system objects that initiate specific processes on the data objects of the index card 2. A trigger may contain a class ID that identifies the object class, a directive that specifies the conditions that activate the trigger, and a method whose execution is initiated by the trigger. The triggers allow for the management of certain specific situations or to implement particular processes relative to the data objects of the index card 2 in a dynamic fashion, i.e. that have not been provided in the basic functions of the card 2. The database 30 may comprise a truth table including the conditions and the actions related thereto.

Referring to figure 2, as depicted by block 50, a trigger 52 can be attached to the master certificate 16 indexed in the memory 22 of the card 2, a predetermined action being executed in relation with the data object 14

associated to the master certificate 16 when the trigger 52 meets a predetermined condition. In addition or alternatively, as depicted by block 54, a trigger 56 can be attached to the data object 14 stored in the data storage medium 12, a predetermined action being executed in relation with the data object 14 and the master certificate 16 when the trigger 56 meets a predetermined condition. The use of triggers 52, 56 not only provides additional functions to the system, but also enables predetermined time-based operations on the data objects in an individual manner without requiring later complex planning and processing.

The secondary certificate 24 may correspond to a cell address on the data storage medium 12 where the data object 14 is stored. In a configuration where the data storage medium 12 comprises information cells managed by a server 8, it may then be indexed on the data storage medium 12, with a reference (logical address) to the information cell of the data storage medium 12 where the data object 14 is stored. The data storage medium 12 is then provided with a concordance table between the secondary certificate 24 and a corresponding memory address in the data storage medium 12.

The data object 14 stored on the data storage medium 12 may have a structure comprising a pointer to an additional data object stored on the data storage medium 12. Such a feature can be useful for example when the data object 14 is an index, or when reorganizing the data objects pertaining to a card.

Referring to figure 1, the system can be designed so that the (index) card 2 is operated in specific ways involving the above-described indexing method. For example, the system may be designed to detect a predetermined condition relative to the memory 22 of the card 2, and if the condition is detected, to apply the indexing method on a number of the data objects 14 stored in the memory 22 of the

card 2. The condition can be for example a saturation of the memory 22 of the card 2, in which case the indexing method is used to free the card's memory 22 to provide space for additional data objects. The detection step can be initiated by the card 2 itself when it is docked in the reader 6, or by an application software in the computer 4 during the processing of the card 2, so that an express request is generated to the system to free the memory 22 of the card 2.

Restrictions on the indexing method can be implemented, so that it is executed, for example, only on the data objects 14 having a mobility property indicating that the data objects 14 are moveable outside the memory 22 of the card 2. This provides additional functionality and flexibility to the system for various applications. In such a case, a mobility property is assigned to each data object 14 based on a character thereof, the mobility property indicating that the data object 14 is moveable outside the memory 22 of the card 2 or not. The locations of the data objects 14 in the system i.e. in a card or in a remote storage medium, can be traced by assigning a location attribute to each data object 14, the location attribute being set to a resident state for each data object 14 stored in the memory 22 of the card 2 and set to a non-resident state for each data object 14 stored outside the memory 22 of the card 2, e.g. in the data storage medium 12 of one of the servers 8.

In addition, an authorized residence period in the card 2 can be assigned to each data object 14 based on the character thereof.

The aforesaid features may be used to determine which and when object data 14 stored in the memory 22 of the card 2 should be moved externally and stored in the storage medium 12, and under which conditions. For example, the indexing method can be applied on every data object 14 having the location attribute set to the resident state, the mobility

property indicating that the data object 14 is moveable, and the residence period elapsed when there remains a predetermined amount of space in the memory of the card 2. It can be applied on a part of any data object 14 having the location attribute set to the resident state, the mobility property indicating that the data object 14 is moveable, and regardless of the residence period when the card 2 is short or plans to be short of free memory space. It can also be applied on the whole of any data object 14 having the location attribute set to the resident state regardless of the residence period when the card 2 is short of memory to carry out an operation. Other conditions may be implemented for automatic processing of the data objects as desired.

A use attribute can be assigned to each data object 14, the use attribute being set to an active state when the data object 14 is solicited, an inactive state when the data object 14 remains unsolicited for a predetermined time period, and an archive state when the data object 14 is archived on the data storage medium 12. The use attribute can be particularly useful in deciding whether a data object should be archived or not depending on the degree of use thereof. Hence, a frequently requested data object should possibly be kept in the memory 22 of the card 2 for fast access, while an infrequently requested or disused data object should possibly be stored externally to leave space for more frequently requested data objects, thereby increasing the efficiency of the card processing. Thus, each data object having the location attribute set to the non-resident state and the use attribute set to the inactive state for a predetermined inactive time period is preferably archived on the data storage medium 12. The attribute can be used by the software application for the automatic repatriation of the data objects 14 from the card's memory 22 or the database 74, 76, 78.

The memory 22 of the card 2 may be structured so that it comprises non-resident object indexes 58 for storing the master certificates 16 of the data objects 14 stored on the data storage medium 12, one or several resident object indexes 60 for storing the master certificates 16 (references) related to the data objects 14 having the location attribute set to the resident state yet that are or have been subjected to a forced indexing, a master index 62 for storing a unique key identifier of the card 2, and a class index 64 for storing a list of classes of the data objects subjected to a forced indexing. The memory 22 of the card 2 may also comprise an index 66 for storing references to nominative files located in remote sites.

The non-resident object indexes 58 may be considered as data objects 14 to which the indexing method is also applicable. Likewise, each master certificate 16 may be a data object 14 to which the indexing method is also applicable.

The IC portable device may be provided with an object dictionary 68 containing object class definitions including directives and methods for the data objects 14. Each data object 14 may then be classified in relation with the definitions in the object dictionary 68. The class definitions may include a priority qualifier determining a data object indexation priority order for applying the indexing method.

The functional blocks shown in figure 1 represent functional devices of the system for carrying out the indexing method according to the invention, and the arrows and rings indicate where the devices can be located as the system may take many configurations. For example, the device executing the indexing, namely the indexing device 70, can be integrated entirely into the operating system (OS) of the smart card 2 or, alternatively, entirely in the reader 6 (or

a SAM) or in the memory 72 of the computer 4, or distributed through these devices. The advantage of integrating the indexing device 70 in the OS of the smart card 2 is that it allows for the secure processing (management of the anonymous references) associated to the master certificates and the transactions. With the master certificates so remaining always in the card 2, a third party (a person or a computer system) is prevented from catching and using them. The indexing device 70 fulfils mainly two functions: the management of non-resident data objects, i.e. those whose displacement to a remote site is predetermined; and the virtual memory generation for the card 2, i.e. the forced displacement of objects that normally reside by definition in the card 2. The card's OS decides this displacement. If the reader 6 is provided with a SAM, the indexing device 70 can be integrated into a software layer of the SAM. With a SAM, the OS of the card 2 is distributed in a client/server fashion, the server portion residing in the SAM while the client portion residing in the card 2. The indexing device 70 can even be embodied in the computer 4, so long as it is secured properly. In such a case, the computer 4 can be used to emulate a virtual IC card.

The database provided by the storage medium 12 can take various forms, like an anonymous database 74, a depersonalized database 76, or a cryptographic database 78. A distinctive feature of the data objects stored in these information warehouses resides in the data being anonymous and optionally encrypted. The personalization of the information (and optionally the decryption), as well as the access control, mainly achieved by the identification of a given data object in the database 74, 76, 78, must necessarily involve the use of a card holding the proper references (certificates).

The indexed objects can be referenced in many ways, using a reference system 80. A reference may correspond to a group of distinct objects (several basic information cells) or gathered together (a cell containing several objects). An open (episodic) reference can also be used, i.e. whose corresponding objects have the property of being capable to evolve or be linked to other objects gathered under the same cell (conversely, it can be said that there are several types of cells in the database).

The card 2 contains indexes that contain the references or certificates, thereby forming an index system 82. An index can be associated to a class of objects or yet be global and containing references to several object classes. Some indexes can be public, i.e. whose content is accessible to the world outside the card 2, while other indexes can be system indexes only accessible by the OS of the card 2.

As previously indicated, a reference generator device 84 generates the references. The references must be unique, random (no links between them), anonymous (no links with the card holder), and have sufficiently long binary sequences to obtain an important referential set to resist to their analysis. The references can be generated using a central random reference generator located in a server 8, or by derivation from a master certificate indexed in the card 2.

The use of the references can be managed by a manager device 86 for this purpose. The references to an object can be handled by different cards that have received authorization to do so by the acquisition of a copy of a reference or a reference derived from the original. The rules of use of the references (aspects of time, function, delegation of rights, transitivity, deletion, limits and scope of the authorizations, etc.) must be handled by a process or a function of the IC card that manages and applies these rules.

The system may be provided with an anonymous backup/restore device 88 that can be invoked whenever a user looses its card or the card becomes defective for any reasons.

The computer 4 executes one or several software applications providing the tools and functions that are relevant to the system's purpose, e.g. a medical application if the kind of data objects to manage with the system is of a medical nature. An example of an already existing software application used in the medical field is Médicarte™, which is a medical software for the management of computerised patient records. In a medical context, the smart card 2 has a double function and a double purpose. It may act as a portable medical record of a patient, and contains medical and administrative data in its memory. In such a case, it is also used so to speak as an access key to the patient record on a server 8. It contains encryption keys in its memory, which allow for the gathering of the content of the patient record and makes it accessible when the patient receives health care services. It may also act as a professional user card, given to a medical practitioner for the management of accesses to the network 10 or to patient records stored in the servers 8. In such a case, the professional user card may hold a digital signature so that the health care professional's name can be linked to each new entry or transaction.

Features found in cryptosystems and networking systems can be implemented in the system according to the invention. For example, authentication processes can be implemented either at the level of the card holder (e.g. password) or the card itself (e.g. digital signature). Mechanisms regarding access rights, non repudiation, data integrity, encryption, are other examples of features that can be implemented in the system.

All data are preferably encrypted before being transferred. A public key cryptosystem can be used for this purpose. The encryption keys are stored in the card's memory (patient or professional card) and should never leave the card. They can therefore neither be copied, nor be intercepted on the network. These encryption keys are used to protect new data entered by a user and that must be transferred on the network 10 and stored in the servers 8. Should a non-authorized person intercept the data during a reverse transfer (from a server's database 16 to a user), he/she will not possess the necessary keys to decrypt the message. A diary of transactions may be used to keep a list of all the activities carried out on the network 10. The corrections made in a patient record, the entry of new data and even the simple consultation of data may be logged in the diary. The physical protection of data at the storage location requires various security mechanisms. Of course, it is recommended to ensure the minimum physical protection of the databases. To complete this protection, the system relies on a special recording system, where all information is recorded anonymously. It is therefore impossible to find the identity of a patient from clinical information. Also, the clinical data of the same patient record is preferably not grouped together in the database 16. Therefore, it does not contain any common identifier that could allow a link to be built up between the data. The patient record is thus in fact a virtual record, which is reconstructed safely when the patient presents his/her card to an authenticated health care professional. In the recording system, all clinical data contained in the patient records may also be automatically copied into a depersonalised database 70. This database does not contain any information on the identity of the patients. On the other hand, the clinical data thus grouped together by patient, makes it useful for statistical analysis on the

population's state of health or for the evaluation of health costs. The system therefore offers to health service managers a vast database that is immediately available at no additional cost, yet which has no effects on the data privacy and security.

Referring to figure 3, there is shown a schematic diagram illustrating a possible architecture of the system in a functional point of view. The anonymous database functions 90 manage the anonymous data warehouse, and provide local security of the data. The emitting functions 92 are for the structuring of the file, the definition of the security of the data and the access rights. The backup database functions 94 manage the backing up and the restoring of the references (certificates) of the index cards. The application 96 performs a semantic management of the information. The API (Application Program Interface) 98 sees to the formatting of the application and network data, the network connection, the optimization of the processes (cache, multitasking, etc.), and the security outside the card. The index card's OS 99 sees to the data access control, the index and reference management, the indexed data management, the virtual memory management, the shared data control, the critical data security, the data anonymity management and the data backup production.

Figure 4 is a diagram showing an example of object classes for the index card IC. The objects of the index card IC are grouped in two metaclasses: the system objects SYS_OBJ and the public objects PUB_OBJ. The public objects are accessible to the world outside the index card IC. These are generally the data objects accessible to the normal actors in the system (e.g. patient, physician). The system objects are objects that are not accessible to the world outside the card IC. These objects are under the unique management (creation, update, etc.) of the card's OS. The objects of the card IC

may have a visibility property that is different whether the object is a public or system object. The visible public objects have no operation restrictions (read, write, etc.) insofar as the requesting party has the proper access rights for the objects. The non-visible public objects are not accessible by the normal actors in the system. These objects remain in the card IC and are considered by the card IC as being non existent. The visibility property is controlled by the actors owning the pre-established authorizations. The visible system objects are management objects accessible in read-only mode to the system's actors, insofar as they have the proper access rights. For example, the dictionary of the card's objects is a visible object. The non-visible system objects are management objects that are not accessible to the world outside the card IC. For example, the data structures used by the card IC for the management of the card's virtual memory is a non-visible system object.

The public objects are divided into two classes: the data objects DATA_OBJ and the management objects MANAG_OBJ. The data objects are subdivided into three classes: the resident objects RES_OBJ, the non-resident objects NR_OBJ and the hybrid objects HYB_OBJ. A resident object is stored in the card IC while a non-resident object is stored outside the card IC, on an external storage medium. Some objects may intrinsically be never written in the card IC, like those whose size exceeds the card's memory capacity. The definition of the objects belonging to these three classes depends on the application field. These objects are formed of primary or complex objects. A primary object is a data atom and it does not contain any other object. A complex object is formed of several primary or complex objects. For a medical card, objects like as follows could be found: objects for the management of the diagnostics DIAG_MED, prescriptions PRES_MED, laboratory examinations BLOOD_FORM, URINE_EXAM,

etc. The management objects cover a set of public objects predefined for the data management. There can be found, among other things, DATE_OBJ for dating transactions, SIGNATURE_OBJ for signing transactions, VER_OBJ for indicating the version of an object, CLASS_ID for identifying an object class, etc.

The system objects are all management objects. Among them, there can be found objects for the management of indexed objects INDX_OBJ, non-resident object indexes INDX_NR_OBJ, a resident object index INDX_VIRT_RES_OBJ, a class index INDX_CLASS_OBJ and a master index INDX_VIRT_MASTER_OBJ. There can be found also an object dictionary DICT_OBJ and a trigger management object TRIGGER_OBJ.

Resident objects are complex data objects of the public domain. They belong to the RES_OBJ class and are stored on the card IC. These objects are generally dynamic, i.e. they can be moved on an external memory to provide the space needed by the card IC. Their indexing priority is defined in the object dictionary DICT_OBJ.

A resident object contains the following primary or complex objects: PRIM_OBJ which is a primary object and/or COMP_OBJ which is a complex object containing the user data (a resident object may contain several objects); DATE_OBJ which is the recording date of the resident object; and SIGNATURE_OBJ which is the electronic signature of the party who stored the resident object.

From the standpoint of the user, the non-resident objects are public objects belonging to the NR_OBJ class. They are complex or primary data objects stored in an external memory of the index card IC. It can be any type of memory (see figure 1). In every case, the public object is encapsulated in a system object belonging to the class SYS_NR_OBJ.

A non-resident object stored in an anonymous database BDA 74 (as shown in figure 1) is encapsulated in a system object of the SYS_NR_BDA class. It contains the following objects: CERTIFICAT_SELEC which is an object referring to an indexed object (it is a selective certificate which allows for the selection of a particular object); and CRYPT_OBJ which is an object containing a cryptogram. By removing the cryptographic capsule, a public complex object of the class NR_OBJ is obtained. This object is formed of the following objects: CLASS_ID which is a public primary object identifying the class to which the non-resident object belongs (subclass of NR_OBJ); VER_OBJ which is a public primary object identifying the object's version; PRIM_OBJ which is a primary object and/or COMP_OBJ which is a complex object that contains the user data; DATE_OBJ which is the recording date of the object; and SIGNATURE_OBJ which is the electronic signature of the party who stored the object in the database.

The non-resident objects of a depersonalised database BDD 76 (as shown in figure 1) are, originally, defined and managed in the same way as the objects of an anonymous database BDA. However, there are differences at the level of the object encapsulation that belongs to the SYS_NR_BDD class. Indeed, unlike objects of a BDA, the certificate is a primary object belonging to the class CERTIFICAT_IC. The certificate is the same for all the objects belonging to a same holder. There are thus no selective certificates. Indeed, the primary goal of a BDD is to allow for the anonymous exploitation of data in an administrative, statistical or search point of view.

The non-resident objects of a cryptographic database BDC 78 (as shown in figure 1) are, somewhat, an aggregation of the system objects of a BDA and of a BDD. Indeed, the encapsulation object belonging to the SYS_NR_BDC contains a

certificate of the class CERTIFICAT_IC, that refers to the entire data of a card as well as a selective certificate of the class CERTIFICAT_SELEC allowing a particular object to be located among the objects of a holder. The CERTIFICAT_IC allows for the retrieving of all the objects of a card, while the CERTIFICAT_SELEC allows for the selection of a specific object.

The non-resident object index is a dynamic system management object. Indeed, the size of the indexes may increase importantly as a function of the transactions achieved. It can thus be screened or simply entirely moved on an external memory. The non-resident object index belongs to the class INDX_OBJ_NR and contains references to data objects residing in the external memory. It may be a BDA or BDC type of memory. The BDD type of memory is not used for transactional operations; there is thus no index referring to specific objects. The index card IC yet keeps the global reference CERTIFICAT_IC to all of its BDD objects. In the case of a non-resident object index referring to objects on a BDA, the index belongs to the class INDX_OBJ_NR_BDA and contains the following objects: CERTIFICAT_SELEC which is the reference of the indexed object on the BDA (it is a selective certificate); DESCRIP_INDX which is a possibly complex object of the public class RES_OBJ, which corresponds to the descriptors of the transaction (this object belongs to a resident class since it is saved in the index card); ADDRESS_XMEM which is the logical address of the BDA; DATE_OBJ which is the creation date of the non-resident object certificate; and SIGNATURE_OBJ which is the electronic signature of the party from who the non-resident object certificate originates.

Every non-resident object index can be merged into a single one, taking into account the access rights to the various objects.

A non-resident object index referring to objects of a BDC is an object belonging to the class INDX_OBJ_NR. It thus contains the same basic objects as an index referring to a BDA. However, there are differences at the referential level. Indeed, a BDC uses two reference certificates: the generic referential of the index card, CERTIFICAT_IC, and a selective certificate of the class CERTIFICAT_SELEC. To manage the objects of a BDC, the index card IC thus uses a complex object INDX_OBJ_NR that contains the following objects: DESCRIP which is a primary or complex object of the public class; RES_OBJ which corresponds to the descriptors of the transaction (formal data); ADDRESS_XMEM which is the logical address of the BDC; DATE_OBJ which is the creation date of the non-resident object certificate; SIGNATURE which is the electronic signature of the party from who the non-resident object certificate originates; and the object INDX_IC which contains the primary object CERTIFICAT_IC which is the card's reference.

The index of the virtual card memory (VCM) is a static system management object. It is an index of dynamic resident objects. It contains the references to objects that are stored in the card IC, but that have been subjected to a forced indexing (mechanism of the card IC activated to provide memory space in certain circumstances). The referenced objects can originate from non-resident object indexes or be resident data objects. A card IC contains a single VCM index. It contains the following primary objects: CLASS_ID which identifies a class of objects (it may be a subclass of the IC or of the IC class); NBR_OBJ_INDEXES which provides the number of indexed objects for an object class or for the whole IC according to the object CLASS_ID; TYPE_VCM which provides the kind of an indexed object: a class or a class object; ADDRESS_XMEM which is the logical address of the external memory (data warehouse) for reference; and

CERTIFICAT_VCM which is the referential in the external memory.

In the case where the class identifier CLASS_OBJ is a subclass of the card IC, if the object TYPE_VCM defines an object class, NBR_OBJ_INDEXES is set to 1 and the certificate points on the entirely indexed class. However, if TYPE_VCM specifies an object, then NBR_OBJ_INDEXES corresponds to the number of occurrences of indexed objects of the class and CERTIFICAT refers to a subset of objects of the class.

In the case where the class identifier is the object IC, NBR_OBJ_INDEXES then indicates the number of indexed classes and the certificate refers to a set of object classes. The number of referred classes is indicated by the object NBR_OBJ_INDEXES. In this case, the card's OS uses the object INDX_CLASS to recognize the identity of the indexed classes.

The card's master index is a static system object belonging to the class INDX_IC. It is the highest-level index. Its scope covers the whole card IC. It contains the object CERTIFICAT_IC that identifies anonymously an index card IC. It is used, among other things, as a referential on a BDD and a BDC.

The class index INDX_CLASS is a static system object. It contains the list of the object classes subjected to a forced indexing. It is constituted by the primary object CLASS_ID that identifies an indexed object class.

The nominative site reference index is a non-mobile (static) public object belonging to the class INDX_NOMIN. This index is different from the other card's indexes. Indeed, it is first a public object and not a system object, which means that it is managed at the application level. It refers to nominative files located in remote sites. Due to his nominative character, it is allotted a static object property, which means that it is never indexed outside the card IC. In the medical field, this index is formed by the

patient index INDEX_PATIENT. It contains the following objects: SITE which identifies a point of service where a nominative file is located; SUPPORT_DOS which identifies the medium on which the file is recorded, i.e. paper or electronic (CD-ROM, hard disk, magnetic tape, etc.); FILE_ID which is the file identifier; and ADDRESS which is an access reference to the file, e.g. telephone number, addresses, etc.

The card's object dictionary is a static system object belonging to the class DICT_OBJ. It contains the description of the object classes existing in the card. It is formed of the following objects: CLASS_ID which is a primary object identifying an object class; PROP_OBJ which is a complex object specifying the properties of the objects of the class, e.g. domain, visibility, residence (number of days of residence), mobility (indexing priority), etc.; INC_OBJ which is a complex object indicating if the objects of the class are indexes, resident objects or non-resident objects and, depending on the case, identifies the links to be established with other objects.

Triggers are static system objects belonging to the class TRIGGER_OBJ. The triggers launch process methods on the card's objects. A trigger contains the following objects: CLASS_ID, which identifies an object class; DIRECTIVE, which specifies the triggering conditions of the trigger; and METHOD, which is the method, launched by the trigger.

The triggers allow to manage certain specific situations or to implement particular processes on the card's objects in a dynamic fashion, i.e. which are not intended in the basic functionality of the card IC.

Referring to figure 1, the indexing process may be carried out in a three-level fashion. The first indexing level of the card 2 is initiated by the non-resident object indexes 58. These indexes 58 contain the selective certificates that refer to objects stored in the external

memory 12. Whatever is the type of the external memory 12, only the selective pointer is used at a transactional level.

The second indexing level is a mechanism that allows the card 2 to achieve an automatic screening of the data objects so as to keep only the most recent (or active) ones in the internal memory of the card 2 and, in this way, to provide the additional memory space in the card 2. The card 2 triggers the indexing when an object exceeds its number of authorized residence days as indicated in the object dictionary 68. In the case where a data screening is effective on all the objects of the card 2 and that the need of space is still present, the card 2 may index active objects on its external memory 12. It is then a forced indexing. It is a first level of virtual memory generation.

To meet the memory space needs, the card 2 has a more drastic means: the integral indexing of the objects. The objects are then entirely indexed on the external memory 12 of the card 2. It is a third indexing level and a second level of virtual memory generation that allows the card 2 to allocate additional space.

The card 2 can only index resident objects having the property of being mobile (dynamic), as defined in the object dictionary 68. The index card 2 takes the priority of an object into account (defined in the object dictionary 68) to determine the data indexing order. The priority may determine the relative importance of an object with respect to other objects of the card 2 or be qualified as a function of other strategies. The card can also apply different strategies based, for example, on an empirical analysis of the rate of use of the objects.

The static objects cannot be moved on the external memory 12. Thus, the more the card's objects are qualified as dynamic, the more the card 2 has the potential for generating virtual memory. However, the card 2 must manage the memory

efficiently to avoid too frequent accesses to the external memory 12 that could slow the system.

Referring to figure 5, there is shown a flow chart illustrating an example of a general process executed by the system with the indexing process as herein above described, for reading and writing operations. Additional operations of various types can be of course also implemented.

As depicted by the block 100, a request is generated by the software application in the computer 4 (shown in figure 1). As depicted by the block 102, the request is transmitted to the reader 6, according to a pre-established protocol, which redirects it to the operating system (OS) of the card 2 as depicted by the block 104. The card 2 can receive a set of requests, like reading and writing requests, as depicted by the block 106. As depicted by the block 108, the card 2 determines the type of the requested operation, and performs the corresponding procedure, like the one for writing a data object as depicted by the block 110 or the one for reading a data object as depicted by the block 112. In the case of a writing operation, the application receives an acknowledgement once the procedure is completed, as depicted by the block 114. In the case of a reading operation, the application receives the resulting data object, as depicted by the block 116. The application then determines whether the data object is an index as depicted by the block 118, and if so, verifies whether the application needs the indexed data objects as depicted by the block 120. In such a case, then the data objects to be retrieved from the memory are selected as depicted by the block 122, and a new request is prepared for the retrieval of the information (block 100). In the case where the data object is not an index or that the application does not need a data object in an index, then the data object can be processed by the application as depicted by the block 124.

Referring to figure 6, a writing request may involve the writing of data objects in the card 2 (resident data objects), in which case the process branches on the right-hand side of the block 124. Otherwise, the procedure for writing data objects on a remote database 12 is initiated, as depicted by the block 126.

Referring to figure 7, the writing of non-resident data objects on a remote database 12 involves the creation of an index and a data entry (certificate address of the database, etc.) for this index in the card 2, as depicted by the block 128. The creation and the management of the indexes are under the unique control of the card's OS, not the application.

Referring back to figure 6, a lack of memory space resulting from the writing request of an object in the card 2 may trigger a procedure for screening the objects in the card 2, for forcing transfer of the objects in the card 2 to the database 12, or for directly indexing the data object which is the subject of the request into the database, as depicted by the steps on the left-hand side branch of the block 130. The incompatibility of the size of an object with the free memory space of the card 2 may also cause the indexing of the object on the database 12, as depicted by the steps on the left-hand side branch of the block 132. Any indexing of data objects on the database 12 must take the mobility property of the objects into account. An object defined as being non mobile cannot be indexed. The management of the virtual memory is achieved through the system index INDX_VCM as shown in figure 4. Of course, other types of implementations are possible.

Referring to figure 8, there is shown representative steps that can be executed in a screening process depicted by the block 134 in figure 6.

Referring to figure 9, a reading request may involve the reading of a resident object, in which case the process

branches on the right-hand side of the block 136. A resident object can be a final data object or an applicative index (index accessible to applications).

Referring to figure 10, during the reading of an index, the random anonymous references (master certificates) are replaced by sequential numbers to maintain their confidentiality, as depicted by the block 138. Likewise, the addresses of the database 12 are masked, as depicted by the block 140.

For reading a non-resident object, the applications must first read the corresponding index as depicted by the block 142 and then, by means of the sequential numbers contained in the index entries, select the indexed data objects to be retrieved (step 122 of figure 5). This process corresponds to a specific and descriptive implementation of the indexing device 70. Other types of implementations are possible.

Figure 11 shows representative steps involved in a process for retrieving an object in a database 12, as depicted by the block 144 of figure 10.

Referring to figure 12, there is shown a schematic diagram illustrating an initial state of the memory card 2 and its virtual memory index 150, and the external auxiliary memory 12 with the database 74.

Referring to figure 13, there is shown a schematic diagram illustrating a writing operation for a resident data object, when the card 2 has enough free memory space. The application 152 transmits a writing request 154 to the card 2 that simply stores the data object 14 in its memory 22.

Referring to figure 14, there is shown a schematic diagram for a writing operation when the card 2 has not enough memory space and the object can be indexed. The application 152 transmits the same request as in figure 13, but the card 2 generates a reference 154 (master certificate) and stores it in the virtual memory index 150. The data

object 14 is transmitted to the external auxiliary memory 12 and stored in the database 74.

Referring to figure 15, there is shown a schematic diagram illustrating a writing operation for a non-resident data object. The application 152 transmits a writing request 156 in this respect to the card 2 that generates a reference 154 (master certificate) for the object and stores it in an applicative index 160. The data object 14 is then stored in the database 74 of the external auxiliary memory 12 by the card 2.

Referring to figure 16, there is shown a schematic diagram illustrating a writing operation when the card 2 has not enough memory space and the data objects in the card 2 can be indexed. As a result, the data objects in the card 2 are screened and sent to the database 74 of the external auxiliary memory 12. This process recovers memory space in the card 2 to allow for example the execution of a requested operation. Upon the request 156 from the application 152, portions of the virtual memory index 158, the applicative index 160 and the resident object 162 are transferred and stored into the database 74 of the external auxiliary memory 12. References 163 (master certificates) to the transferred portions are stored in the resident part of the virtual memory index 150.

Referring to figure 17, there is shown a schematic diagram illustrating a process similar to the process of figure 16, where the objects and the applicative index are transferred entirely to the database 74 and indexed in the virtual memory index 150 of the card 2 using a single master certificate 165.

Referring to figure 18, there is shown a schematic diagram illustrating a reading operation when the requested object 14 is not an index and is located in the card 2. As a result, the object 14 is simply transmitted to the

application 152 in response to the request 158 in this respect.

Referring to figure 19, there is shown a schematic diagram illustrating a reading operation when the requested object is an index 160 and is located in the card 2. As a result, the references (master certificates) in the applicative index 160 are replaced by sequential numbers and the database addresses are masked. The card 2 transmits thus a modified applicative index 168 to the application 152 in response to the request 162 in this respect.

Referring to figure 20, there is shown a schematic diagram illustrating a reading operation where an indexed non-resident object 170 is identified by the information providing from an index entry 164 in the applicative index 160 located in the card 2. As a result, the object 170 is found in the database 74 from the index entry 164 transmitted from the application 152 to the card 2, and transmitted to the application 152.

Referring to figure 21, there is shown a schematic diagram illustrating a reading operation where an object 14 has been transferred out of the card 2 to the database 74 as a result of a former lack of memory space. The object 14 is found back in the database 74 through the virtual memory management index 150 of the card 2.

The following description provides a typical scenario where the indexing device 70 is used for the management of data objects in a healthcare system, where a patient consults a healthcare practitioner. Referring to figure 1, both parties insert their respective cards in the card reader 6. The healthcare practitioner needs to save information (a data object) in the electronic file of the patient, provided by the card 2. By definition, this object is of a non-resident type and consequently is automatically indexed in the remote database 12. During the process, the system depersonalizes

the data object (and optionally encrypts the object for example insofar as the country regulations allow it), identifies the object by a random anonymous reference provided by a certificate as hereinabove explained, and transmits the (encrypted) object via the network 10 to the database server 8 where the data object is stored among the data objects belonging to other persons. The anonymous certificate generated as a result of this transaction is stored in the card 2 of the patient with a transaction descriptor, for example the type of prescribed laboratory examination "Blood formula of June 2, 1999", the eventual result of the examination being the indexed data, or yet a descriptor defining an image "Identification photo of the patient, April 15, 1999", the indexed data being the photo.

The certificate stored in the card 2 is the unique identifier allowing the data object to be meaningfully retrieved in the database 12. The patient can transfer a copy of the certificate or preferably a certificate derived therefrom to the card (not shown) of a third party as the healthcare practitioner, so as to authorize the practitioner for having access to the data object at a later time, e.g. when the laboratory results will be available. The management of this authorization (the derived certificate) is achieved by a process integrated in the operating system of the card of the practitioner (or the index cards in general unless such a feature is only needed for a given type of users).

Thereafter, the holder of the certificates will have access to the associated data objects as follows (scenario of a simple transaction involving non-resident objects). The application executed by the computer 4 requests the reading of the descriptor of an index to the operating system of the authorized card (OS of a real card or OS of a virtual card). The index contains a set of certificates referring to data objects. The operating system of the card returns the index

information except a number of confidential variables like, among other things, the certificates of the indexed data objects. The application returns the descriptors of the transaction for which the corresponding indexed data objects are required to the operating system of the card. Upon receiving the descriptors, the operating system of the card communicates with the server 8 of the database(s) 12 whose address is specified in the index of the card. In the transaction, the operating system of the card includes the certificates relative to the data objects. These certificates can be encrypted and/or derived to prevent security leaks. On the remote site, the server 8 of the database 12 searches the objects corresponding to the certificates and returns them to the requiring site. The exchange preferably involves the use of security protocols for identifying the involved entities in the transaction along with security protocols associated to the confidentiality, the integrity and non repudiation of the transaction. Once the objects are received, the operating system of the card makes the information readable (decryption, etc.) and returns it to the application.

The following description provides a typical scenario for using the indexing device 70 for the virtual card memory generation for an IC card 2.

In the case where the application requires the storage of data in the card 2 and that the memory space of the card 2 is insufficient, the operating system of the card 2 activates itself the indexing process for generating the virtual memory. The operation is about the same as previously described. The fundamental difference resides in the fact that it is the operating system of the card 2 that takes the initiative of performing a data indexing. In the case where the request is made to the card 2 for obtaining the information (which is supposed to be in the card), the operating system of the card will retrieve the data objects

that it has transferred to the database 12 on the remote site in a transparent manner to the user. The application generates a request for storing an object on the card 2. The operating system of the card 2 verifies the available memory space, determines whether the requested space is insufficient, and verifies whether it is possible to establish a network connection with a remote site (directory search). If no network connection is possible, the transaction is aborted. If a network connection is possible, the operating system of the card 2 verifies the attributes of the objects stored in the card 2 to detect those that it can transfer and selects the objects according to different algorithms, like the less used objects, the largest size objects, etc. Afterwards, the operating system of the card 2 generates or obtains a global reference for the group of selected objects and stores it in a system index in the card 2 (which is not available to the world outside the card 2) with a set of relevant information. The operating system of the card then transforms the data objects in a monolithic binary block and transfers it in a secure fashion toward the database 12 that it has selected. It is somewhat like if the operating system of the card 2 would carry a memory page. It should be noted that neither the certificates nor the address of the selected database 12 (in the case where there are several) are known to the world outside the card (e.g. the applications). When the data objects are requested by the application, the operating system of the card 2 searches in the database 12 in a transparent fashion, and delivers the requested objects to the application.

The operating system of the card 2 can also use another strategy to recover memory space: it can proceed with a screening of the card's objects, i.e. transfer to the database 12 of the occurrences of the object classes that are no longer active, including indexes, e.g. the thirty-day and

older pharmaceutical prescriptions. The process is the same as for the virtual memory generation. In this case, the operating system of the card 2 adds summary information to the index or to the zone of purged data regarding the existence of other occurrences of objects.

In the case where there is no available network, the operating system of the card 2 cannot use the indexing device 70, unless it temporarily saves the information locally, which is not recommended since the availability of the working station is not guaranteed and for security reasons. However, it is possible to temporarily and safely store a transaction in a securised server with a SAM when applicable. Such a server provides a safe link between several points of service and the network servers 8 (BDA, etc.). Likewise, if the network is unavailable, it is not possible to have reading access to remote objects, including memory pages of the virtual memory.

While embodiments of this invention have been illustrated in the accompanying drawings and described above, it will be evident to those skilled in the art that changes and modifications may be made therein without departing from the essence of this invention. All such modifications or variations are believed to be within the scope of the invention as defined by the claims appended hereto.

WHAT IS CLAIMED IS:

1. A method of securely expanding a storage capacity of a memory of an IC portable device, comprising the steps of, for a data object stored or to be stored in the memory of the IC portable device:

randomly generating a unique master certificate used as a descriptor of the data object;

indexing the master certificate in the memory of the IC portable device;

deriving a secondary certificate from the master certificate, the secondary certificate being determinable only using the master certificate;

storing the data object with the secondary certificate on a data storage medium external to the IC portable device;

whereby the memory of the smart card is freed from the data object as the data object is stored on the data storage medium, thereby securely expanding the storage capacity of the memory of the IC portable device as only the IC portable device has key information to retrieve the data object stored on the data storage medium.

2. The method according to claim 1, comprising the additional step of:

attaching a descriptive label to the master certificate, the descriptive label representing a character of the data object;

and wherein:

the descriptive label is indexed with the master certificate in the memory of the IC portable device.

3. The method according to claim 1, comprising the additional step of:

encrypting the data object prior to the storing step, the data object stored on the data storage medium being encrypted.

4. The method according to claim 1, comprising the additional steps of:

randomly generating a data access authorization certificate derived from the master certificate and with which the secondary certificate is determinable; and

storing the authorization certificate in an IC portable device held by a third party,

whereby the data object stored on the data storage medium is retrievable with the IC portable device of the third party.

5. The method according to claim 4, wherein:

the authorization certificate is transmitted to the IC portable device held by the third party via a secured communication channel using a cryptographic protocol.

6. The method according to claim 4, wherein:

the authorization certificate is stored in the IC portable device held by the third party through a secured communication channel with the IC portable device from which the master certificate originates.

7. The method according to claim 4, comprising the additional step of:

attaching a trigger to the authorization certificate, the trigger carrying authorization use instructions for the IC portable device held by the third party.

8. The method according to claim 7, wherein the instructions are selected among a life duration of the authorization certificate, a service location restriction, a third party class restriction, and a transitive authorization control.

9. The method according to claim 1, wherein the certificates are managed by a secured microprocessor card server.

10. The method according to claim 1, wherein the master and secondary certificates are generated using a cryptographic algorithm.

11. The method according to claim 1, comprising, prior to the generating step, the additional step of:

inputting the data object into a secured temporary memory in response to a software application request.

12. The method according to claim 1, wherein:

the IC portable device comprises a database of actions in relation with conditions associated thereto, and triggers that trigger execution of each action for which the associated condition is met.

13. The method according to claim 1, wherein:

the data storage medium is connected to a server comprising a database of actions in relation with conditions associated thereto, and triggers that trigger execution of each action for which the associated condition is met.

14. The method according to claim 13, wherein:

the database comprises a truth table including the conditions and the actions related thereto.

15. The method according to claim 1, comprising the additional step of:

attaching a trigger to the master certificate indexed in the memory of the IC portable device, a predetermined action being executed in relation with the data object associated to the master certificate when the trigger meets a predetermined condition.

16. The method according to claim 1, comprising the additional step of:

attaching a trigger to the data object stored in data storage medium, a predetermined action being executed in relation with the data object and the master certificate when the trigger meets a predetermined condition.

17. The method according to claim 1, wherein:

the secondary certificate corresponds to a cell address on the data storage medium where the data object is stored.

18. The method according to claim 17, wherein:

the data storage medium is provided with a concordance table or a function between the secondary certificate and a corresponding memory address in the data storage medium.

19. The method according to claim 1, wherein:

the data object stored on the data storage medium has a structure comprising a pointer to an additional data object stored on the data storage medium.

20. The method according to claim 1, wherein:

the certificates are generated by the IC portable device.

21. The method according to claim 1, wherein:

the master certificate remains at all time in a secured memory zone distinct from the data storage medium.

22. The method according to claim 21, wherein:

the secured memory zone is in the memory of the IC portable device.

23. The method according to claim 1, wherein:

the IC portable device has a unique key identifier, and the master certificate is generated from the key identifier.

24. The method according to claim 1, wherein:

the data storage medium has a unique reference domain identifier and is provided with a central certificate generator, the master certificate being randomly generated, from the unique reference domain identifier, by the central certificate generator, and transmitted to the IC portable device via a secured link.

25. The method according to claim 1, wherein:

the data storage medium is a data warehouse.

26. The method according to claim 2, wherein the descriptive label relates to a health care type of service.

27. The method according to claim 4, wherein:

the third party is a doctor.

28. A method of operating an IC portable device having a memory for storing data objects, comprising the steps of:

detecting a predetermined condition relative to the memory of the IC portable device; and

if the condition is detected, applying the method according to claim 1 on a number of the data objects stored in the memory of the IC portable device.

29. The method according to claim 28, wherein:

the condition is a saturation level of the memory of the IC portable device.

30. The method according to claim 28, wherein:

the condition is an express request to free the memory of the IC portable device.

31. The method according to claim 28, wherein the step of applying the method according to claim 1 is executed only on the data objects having a mobility property indicating that the data objects are moveable outside the memory of the IC card device.

32. The method according to claim 31, comprising the additional steps of:

assigning the mobility property to each data object based on a character thereof; and

assigning a location attribute to each data object, the location attribute being set to a resident state for each data object stored in the memory of the IC card device and set to a non-resident state for each data object stored outside the memory of the IC card device.

33. The method according to claim 28, comprising the additional steps of:

assigning a mobility property to each data object based on a character thereof, the mobility property indicating that the data object is moveable outside the memory of the IC card device;

assigning a location attribute to each data object, the location attribute being set to a resident state for each data object stored in the memory of the IC card device and set to a non-resident state for each data object stored outside the memory of the IC card device; and

assigning an authorized residence period in the IC portable device to each data object based on the character thereof;

and wherein the step of applying the method according to claim 1 is executed on:

a number of the data objects having the location attribute set to the resident state, the mobility property indicating that the data object is moveable, and the residence period elapsed when there remains a predetermined amount of space in the memory of the IC portable device;

on parts of the data objects having the location attribute set to the resident state, the mobility property indicating that the data object is moveable, and regardless of the residence period when the IC portable device is short or expects to be short of free memory space; and

on whole ones of the data objects having the location attribute set to the resident state regardless of the residence period when the IC portable device is short of memory to carry out an operation.

34. The method according to claim 33, comprising the additional steps of:

assigning a use attribute to each data object, the use attribute being set to an active state when the data object is solicited, an inactive state when the data object remains unsolicited for a predetermined time period, and an archive state when the data object is archived on the data storage medium; and

archiving each data object having the location attribute set to the non-resident state and the use attribute set to the inactive state for a predetermined inactive time period on the data storage medium.

35. The method according to claim 34, wherein:

the memory of the IC portable device comprises non-resident object indexes storing the master certificates of the data objects stored on the data storage medium, a resident object index storing references to the data objects having the location attribute set to the resident state and subjected to a forced indexing, a master index storing a unique key identifier of the IC portable device, and a class index storing a list of classes of the data objects subjected to a forced indexing.

36. The method according to claim 35, wherein:

the memory of the IC portable device comprises an index storing references to nominative files located in remote sites.

37. The method according to claim 35, wherein the non-resident object indexes are data objects to which the method according to claim 1 is also applicable.

38. The method according to claim 28, wherein:

each master certificate is a data object to which the method according to claim 1 is also applicable.

39. The method according to claim 28, wherein:

the IC portable device has an object dictionary containing object class definitions including directives and methods for the data objects;

and the method comprises the additional step of classifying each data object in relation with the definitions in the object dictionary.

40. The method according to claim 39, wherein:

the class definitions include a priority qualifier determining a data object indexation priority order for applying the method according to claim 1.

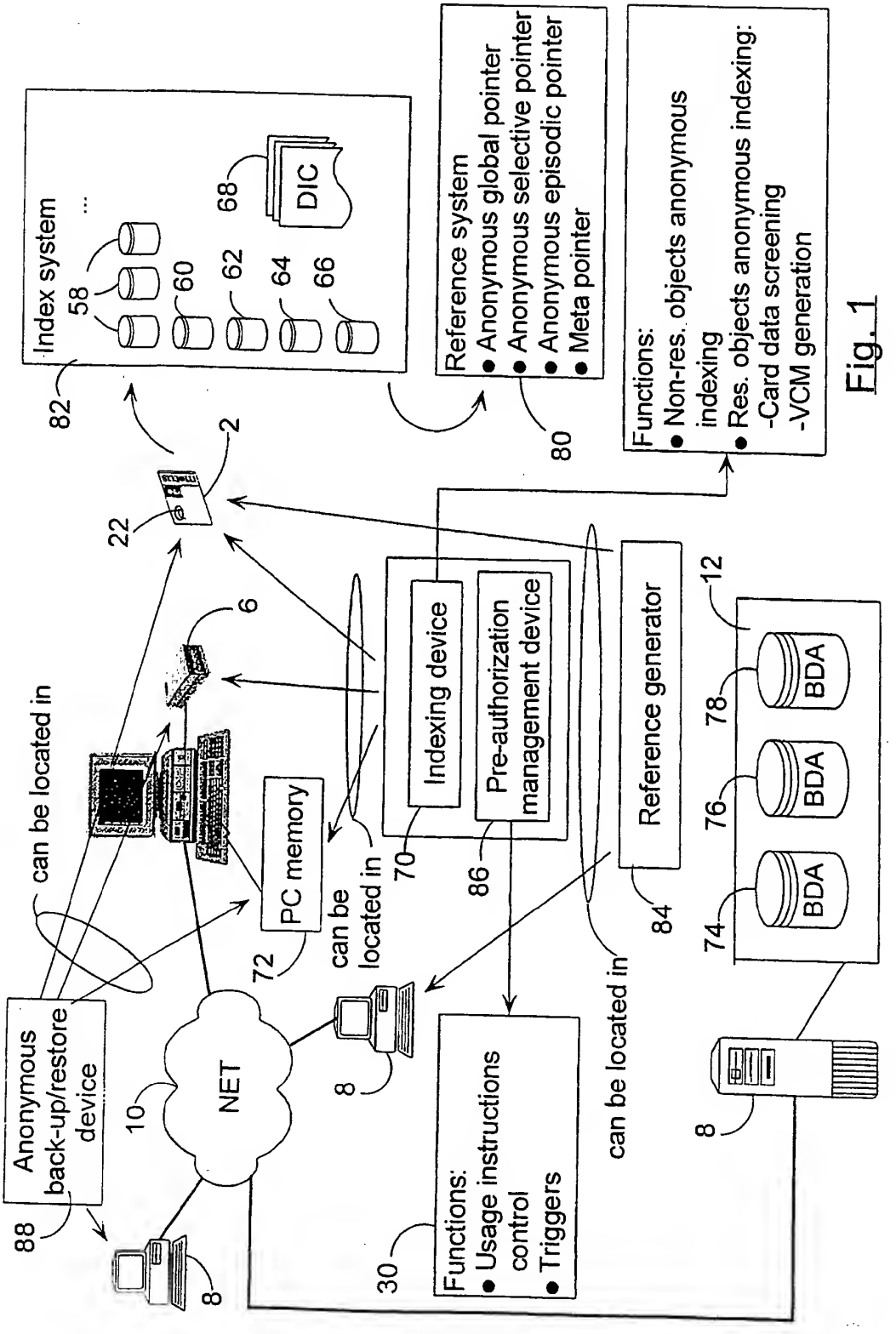
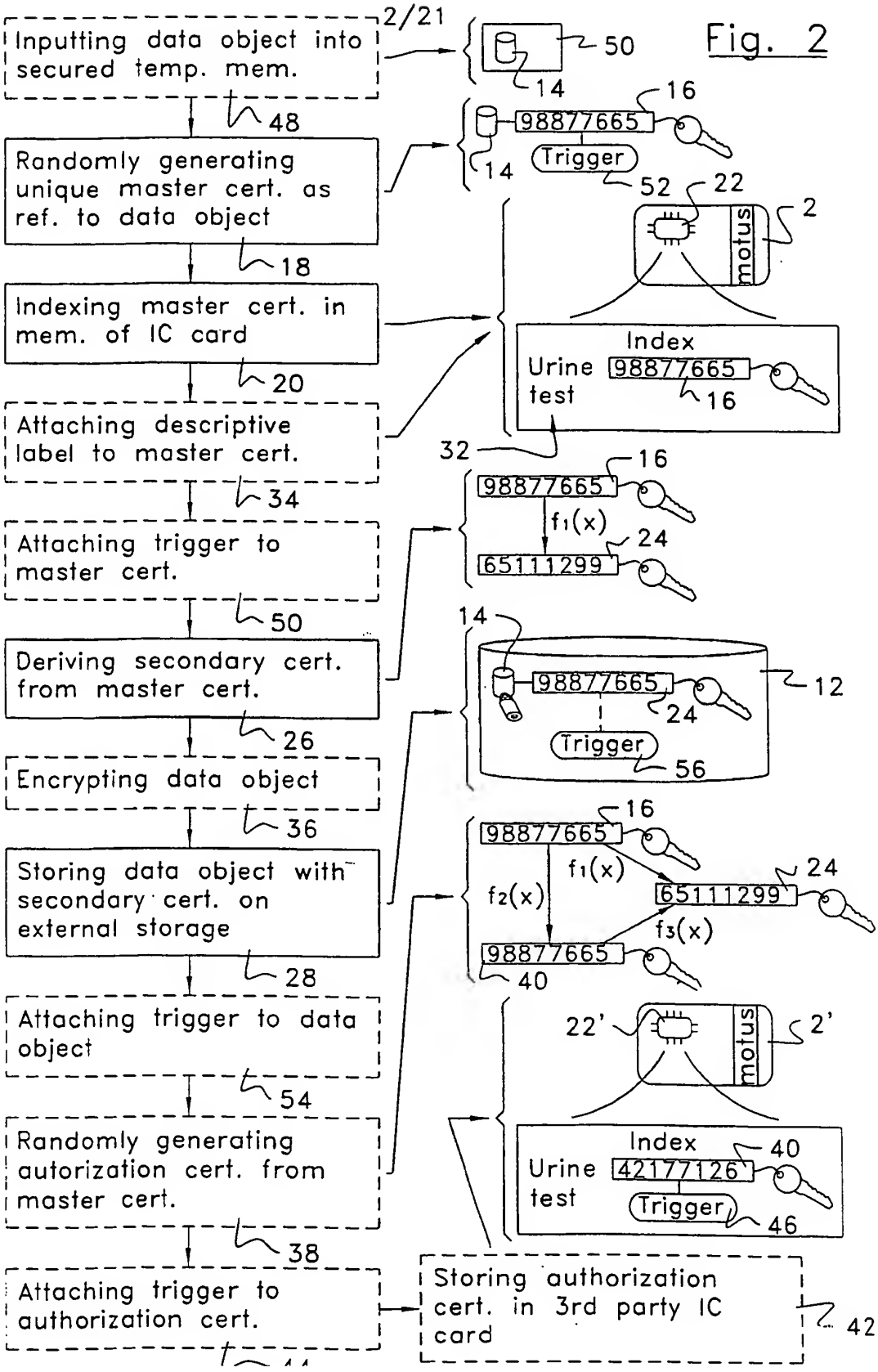

Index system
58
60
62
64
66
68
DIC
82
Reference system
• Anonymous global pointer
• Anonymous selective pointer
• Anonymous episodic pointer
• Meta pointer
80
Functions:
• Non-res. objects anonymous indexing
• Res. objects anonymous indexing:
-Card data screening
-VCM generation
2
22
6
Indexing device
Pre-authorization management device
70
86
Reference generator
84
12
BDA
BDA
BDA
78
76
74
PC memory
72
can be located in
can be located in
can be located in
Anonymous back-up/restore device
88
NET
10
8
8
8
Functions:
• Usage instructions control
• Triggers
30

Fig. 1

Fig. 2


Inputting data object into secured temp. mem.
48
Randomly generating unique master cert. as ref. to data object
18
Indexing master cert. in mem. of IC card
20
Attaching descriptive label to master cert.
34
Attaching trigger to master cert.
50
Deriving secondary cert. from master cert.
26
Encrypting data object
36
Storing data object with secondary cert. on external storage
28
Attaching trigger to data object
54
Randomly generating autorization cert. from master cert.
38
Attaching trigger to authorization cert.
44
Storing authorization cert. in 3rd party IC card
42
50
14
16
98887665
Trigger
52
22
2
motus
Index
Urine test
98887665
16
32
$f_1(x)$
24
65111299
12
56
$f_2(x)$
$f_3(x)$
40
22'
2'
42177126
46

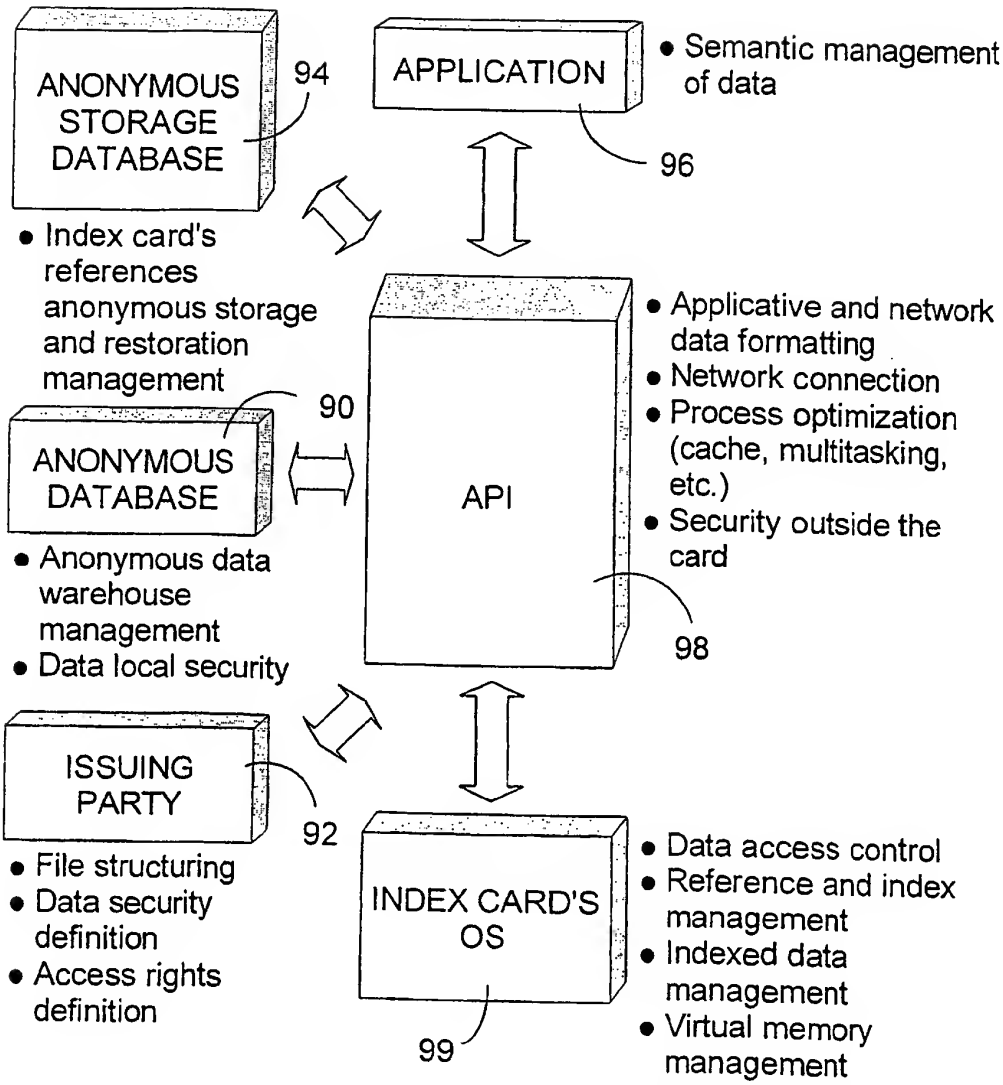

ANONYMOUS STORAGE DATABASE
94
APPLICATION
• Semantic management of data
96
• Index card's references anonymous storage and restoration management
90
ANONYMOUS DATABASE
• Anonymous data warehouse management
• Data local security
API
• Applicative and network data formatting
• Network connection
• Process optimization (cache, multitasking, etc.)
• Security outside the card
98
ISSUING PARTY
92
• File structuring
• Data security definition
• Access rights definition
INDEX CARD'S OS
99
• Data access control
• Reference and index management
• Indexed data management
• Virtual memory management
• Shared data control
• Critical data security
• Data anonymity management
• Storage data production

Fig. 3

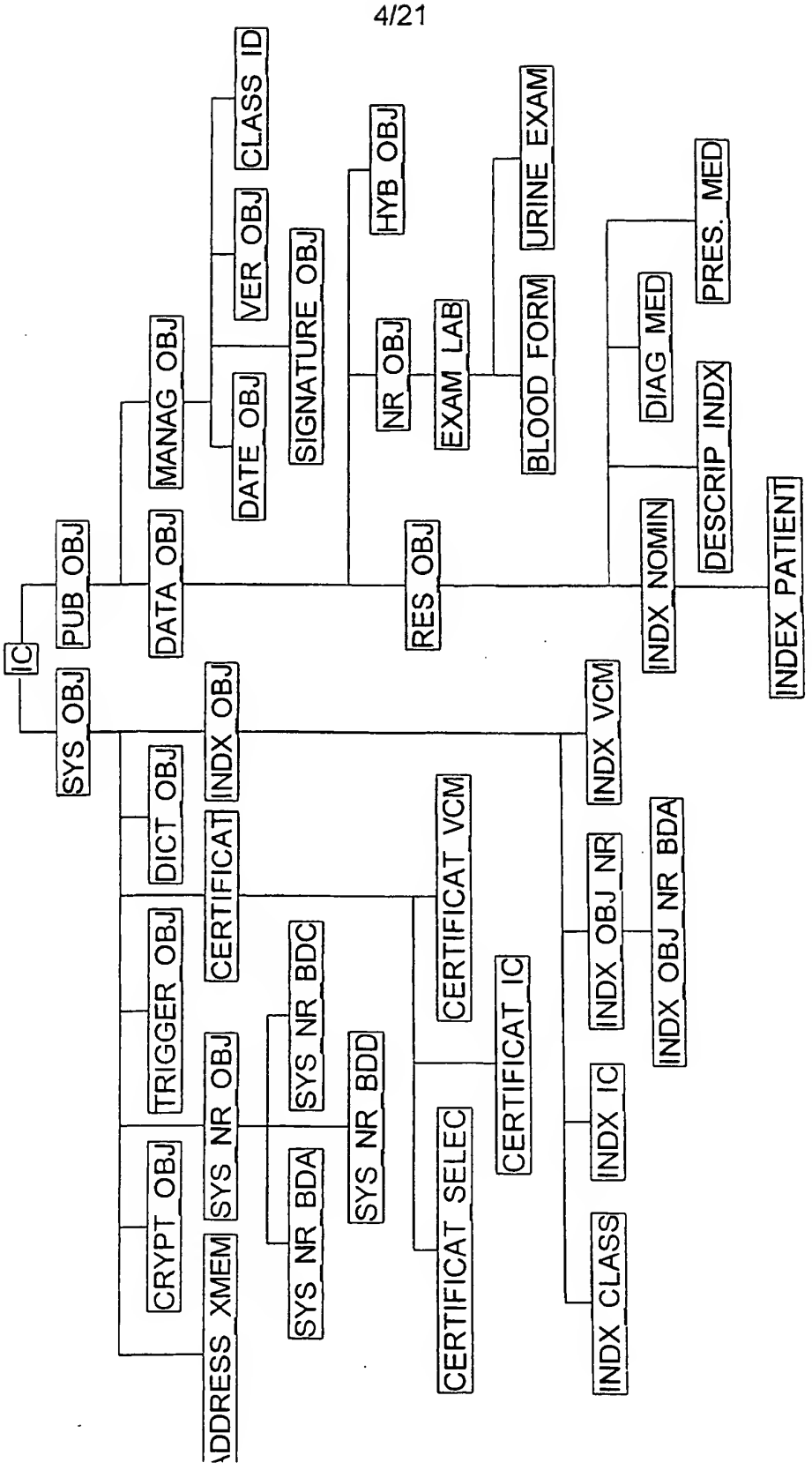

IC
SYS_OBJ
PUB_OBJ
DATA_OBJ
MANAG_OBJ
DATE_OBJ
VER_OBJ
CLASS_ID
SIGNATURE_OBJ
RES_OBJ
NR_OBJ
HYB_OBJ
EXAM_LAB
BLOOD_FORM
URINE_EXAM
INDX_NOMIN
DIAG_MED
PRES._MED
DESCRIP_INDX
INDEX_PATIENT
INDX_OBJ
DICT_OBJ
CERTIFICAT
TRIGGER_OBJ
CRYPT_OBJ
SYS_NR_OBJ
ADDRESS_XMEM
SYS_NR_BDA
SYS_NR_BDC
SYS_NR_BDD
CERTIFICAT_VCM
CERTIFICAT_SELEC
CERTIFICAT_IC
INDX_VCM
INDX_OBJ_NR
INDX_OBJ_NR_BDA
INDX_IC
INDX_CLASS

Fig. 4

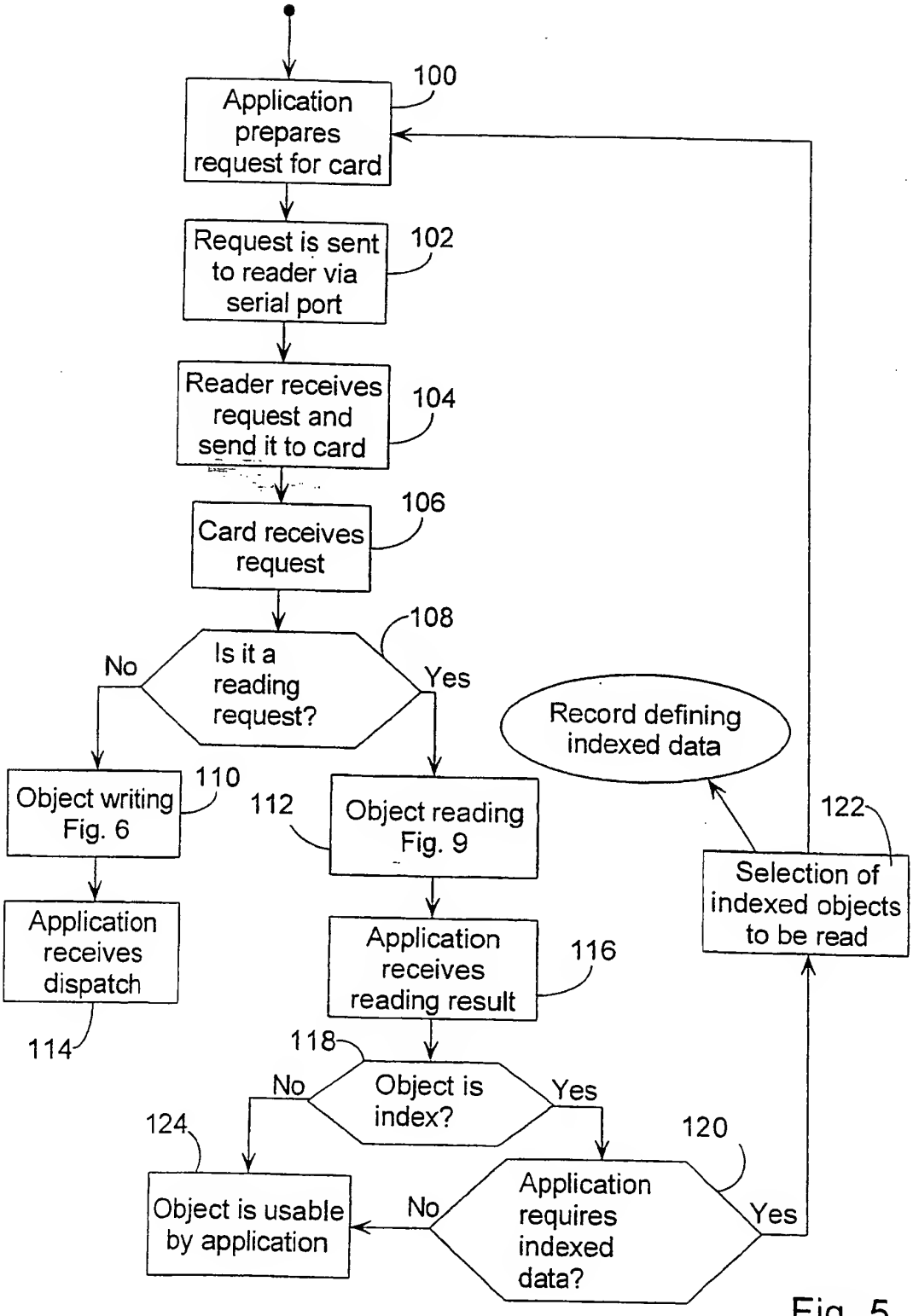

Application prepares request for card
100
Request is sent to reader via serial port
102
Reader receives request and send it to card
104
Card receives request
106
Is it a reading request?
108
No
Yes
Object writing Fig. 6
110
Application receives dispatch
114
Object reading Fig. 9
112
Application receives reading result
116
Object is index?
118
No
Yes
Object is usable by application
124
Application requires indexed data?
120
No
Yes
Record defining indexed data
Selection of indexed objects to be read
122

Fig. 5

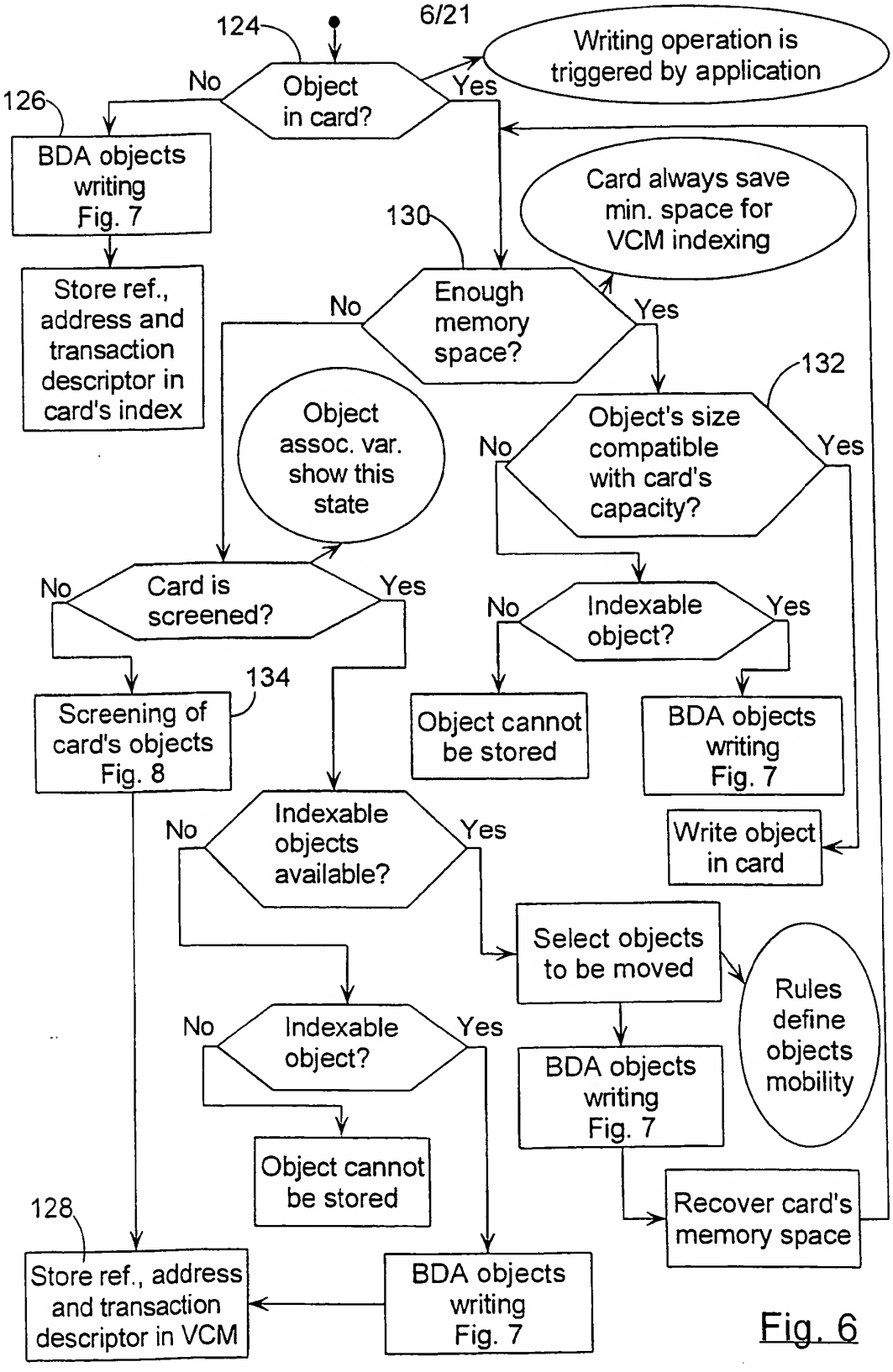

124
Writing operation is triggered by application
No
Object in card?
Yes
126
BDA objects writing Fig. 7
Card always save min. space for VCM indexing
130
Store ref., address and transaction descriptor in card's index
No
Enough memory space?
Yes
132
Object assoc. var. show this state
No
Object's size compatible with card's capacity?
Yes
No
Card is screened?
Yes
No
Indexable object?
Yes
134
Screening of card's objects Fig. 8
Object cannot be stored
BDA objects writing Fig. 7
No
Indexable objects available?
Yes
Write object in card
Select objects to be moved
Rules define objects mobility
No
Indexable object?
Yes
BDA objects writing Fig. 7
Object cannot be stored
Recover card's memory space
128
Store ref., address and transaction descriptor in VCM
BDA objects writing Fig. 7

Fig. 6

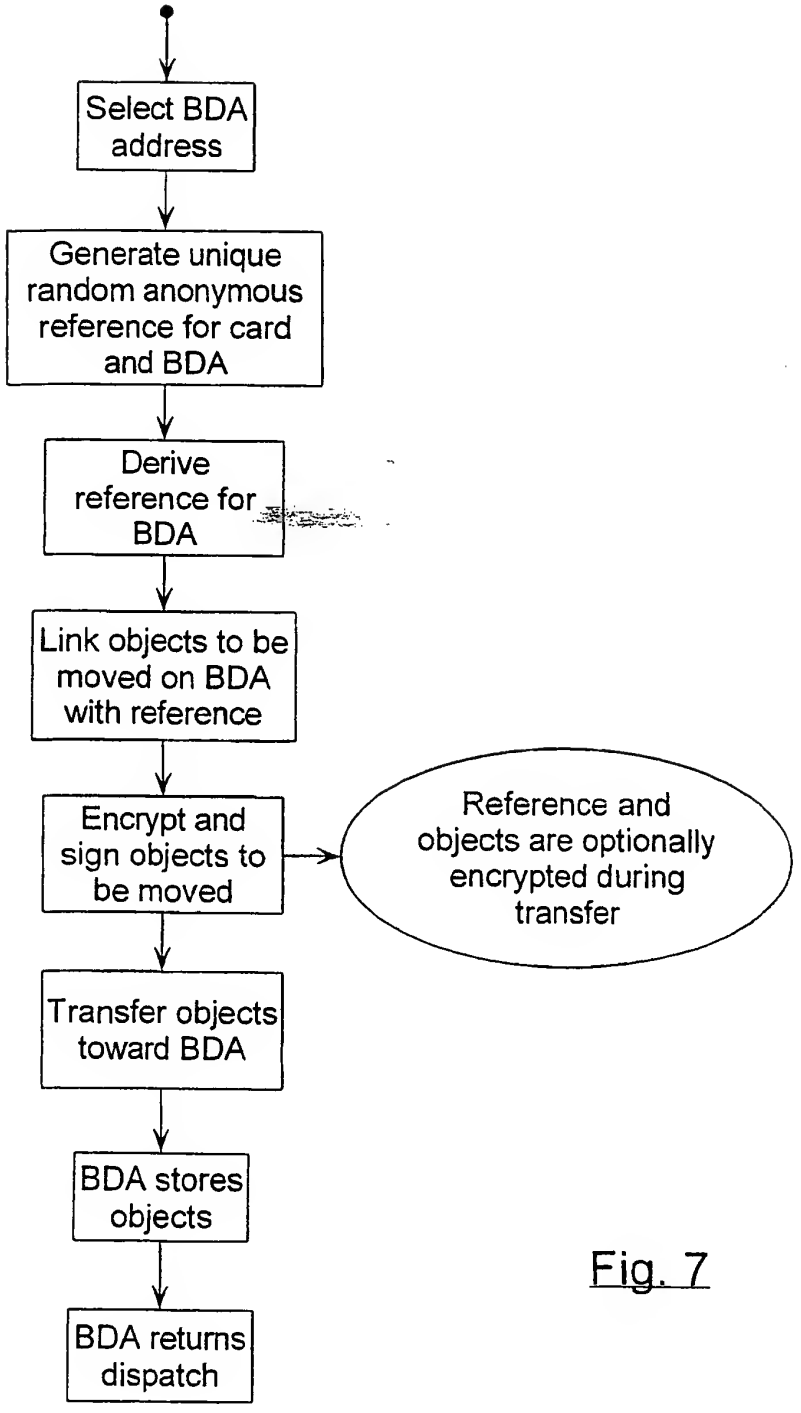

Select BDA address
Generate unique random anonymous reference for card and BDA
Derive reference for BDA
Link objects to be moved on BDA with reference
Encrypt and sign objects to be moved
Reference and objects are optionally encrypted during transfer
Transfer objects toward BDA
BDA stores objects
BDA returns dispatch

Fig. 7

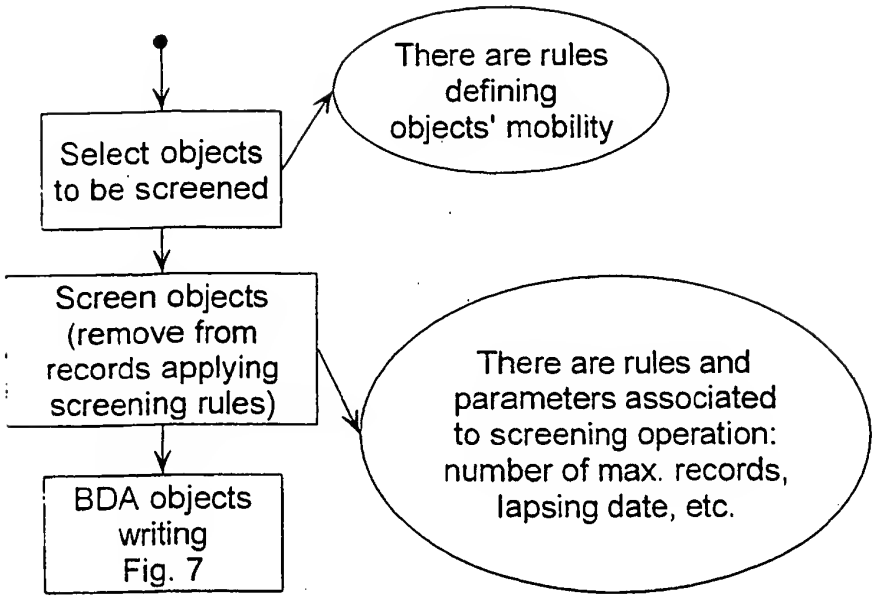
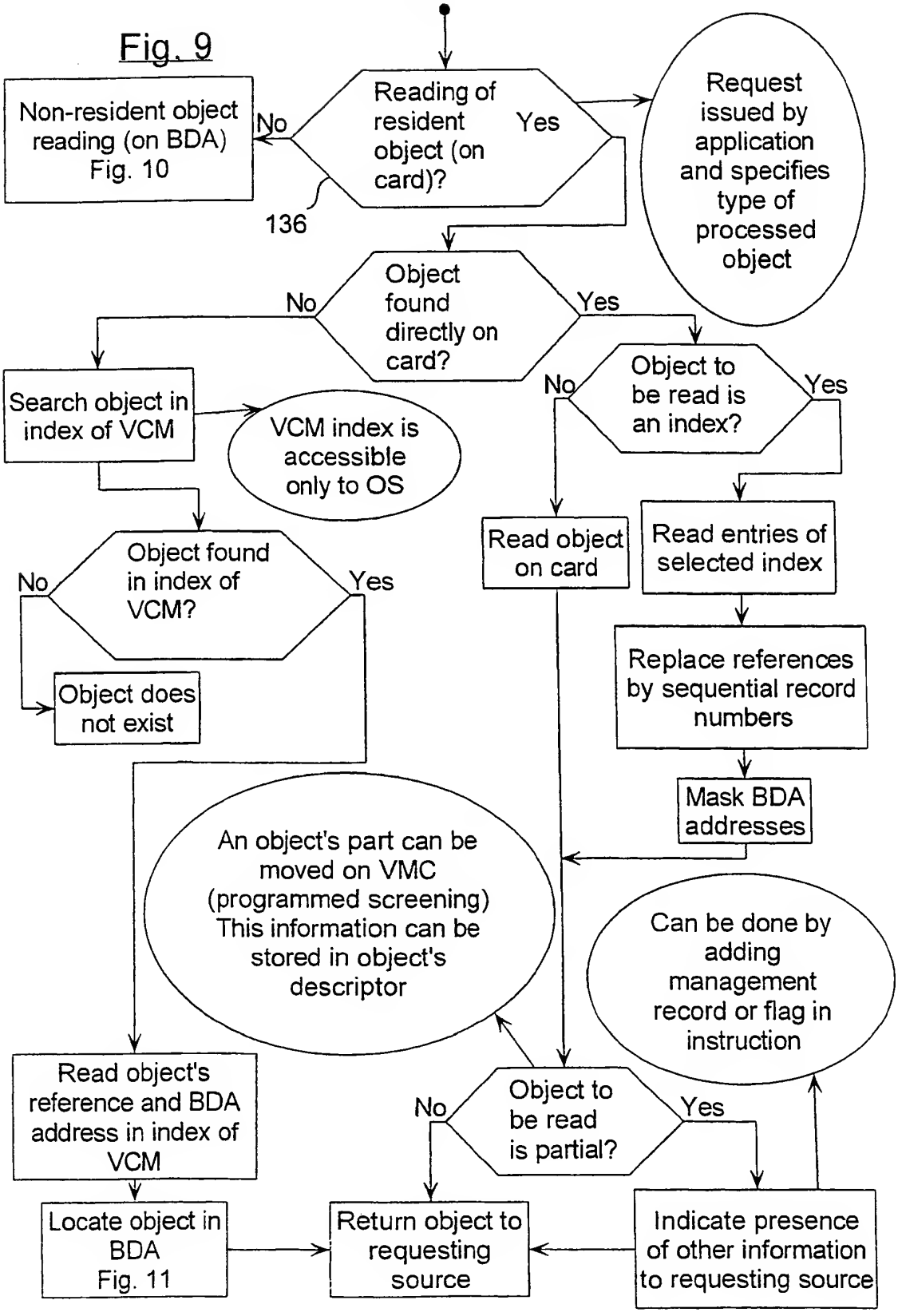

There are rules defining objects' mobility
Select objects to be screened
Screen objects (remove from records applying screening rules)
There are rules and parameters associated to screening operation: number of max. records, lapsing date, etc.
BDA objects writing Fig. 7

Fig. 8

Fig. 9


Non-resident object reading (on BDA) Fig. 10
No
Reading of resident object (on card)?
Yes
136
Request issued by application and specifies type of processed object
No
Object found directly on card?
Yes
Search object in index of VCM
VCM index is accessible only to OS
No
Object to be read is an index?
Yes
Read object on card
Read entries of selected index
No
Object found in index of VCM?
Yes
Object does not exist
Replace references by sequential record numbers
Mask BDA addresses
An object's part can be moved on VMC (programmed screening) This information can be stored in object's descriptor
Can be done by adding management record or flag in instruction
Read object's reference and BDA address in index of VCM
No
Object to be read is partial?
Yes
Locate object in BDA Fig. 11
Return object to requesting source
Indicate presence of other information to requesting source

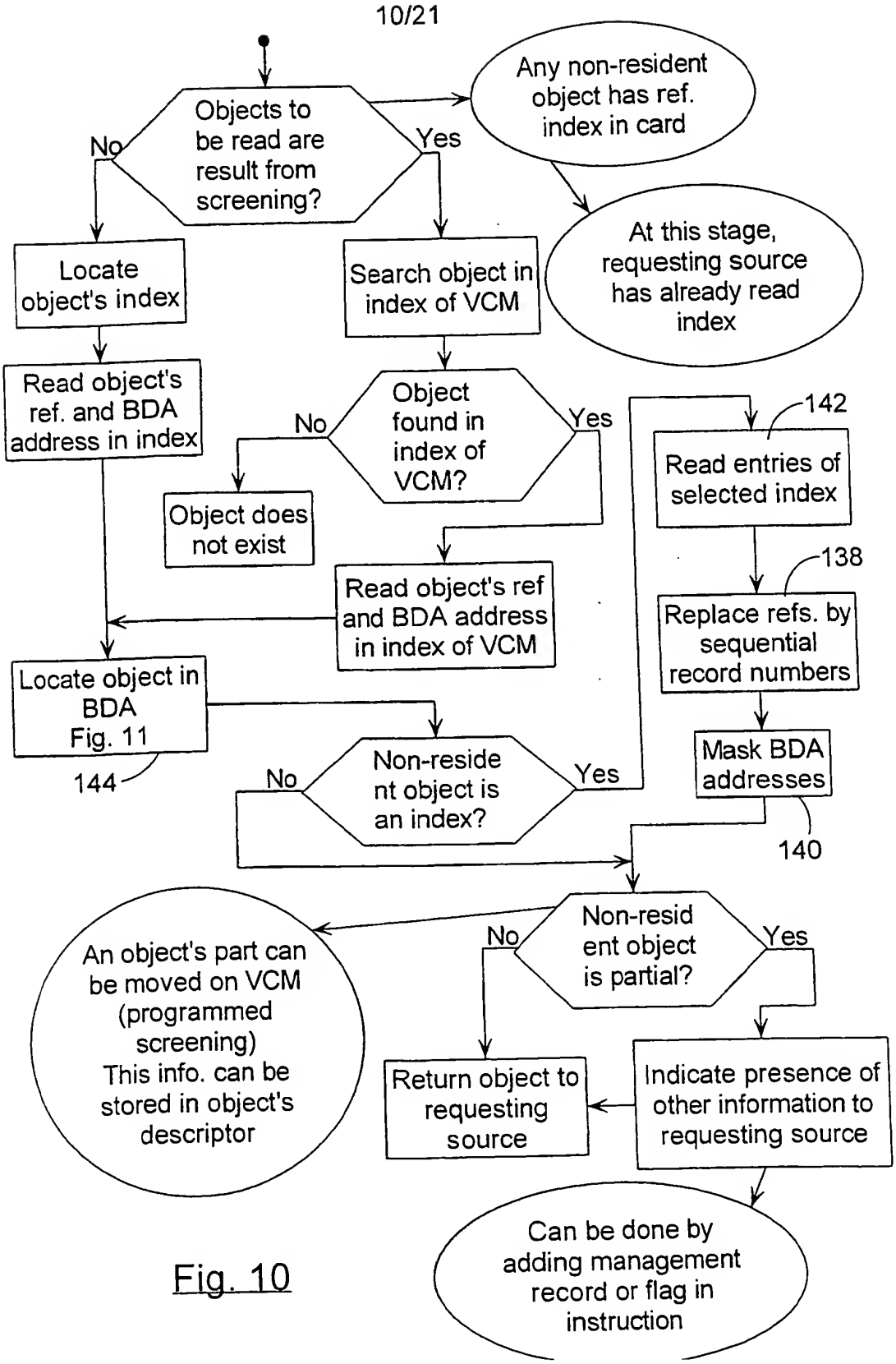

Objects to be read are result from screening?
No
Yes
Any non-resident object has ref. index in card
At this stage, requesting source has already read index
Locate object's index
Search object in index of VCM
Read object's ref. and BDA address in index
Object found in index of VCM?
No
Yes
Object does not exist
Read object's ref and BDA address in index of VCM
Locate object in BDA Fig. 11
144
Non-reside nt object is an index?
No
Yes
142
Read entries of selected index
138
Replace refs. by sequential record numbers
Mask BDA addresses
140
Non-resid ent object is partial?
No
Yes
An object's part can be moved on VCM (programmed screening) This info. can be stored in object's descriptor
Return object to requesting source
Indicate presence of other information to requesting source
Can be done by adding management record or flag in instruction

Fig. 10

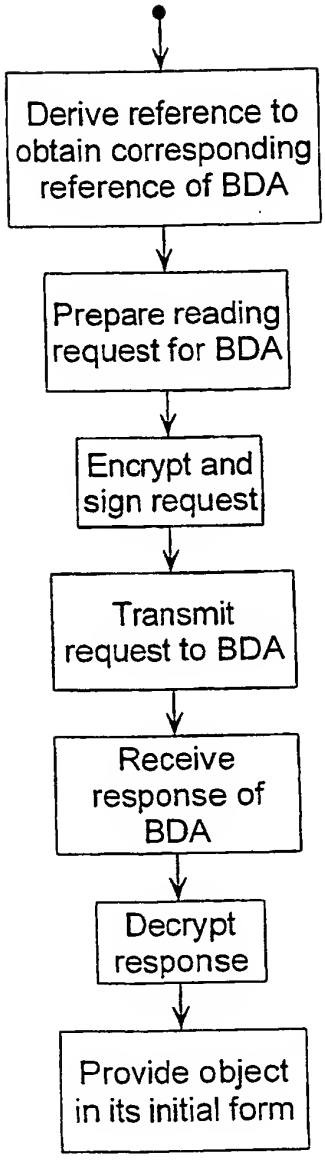

Derive reference to obtain corresponding reference of BDA
Prepare reading request for BDA
Encrypt and sign request
Transmit request to BDA
Receive response of BDA
Decrypt response
Provide object in its initial form

Fig. 11

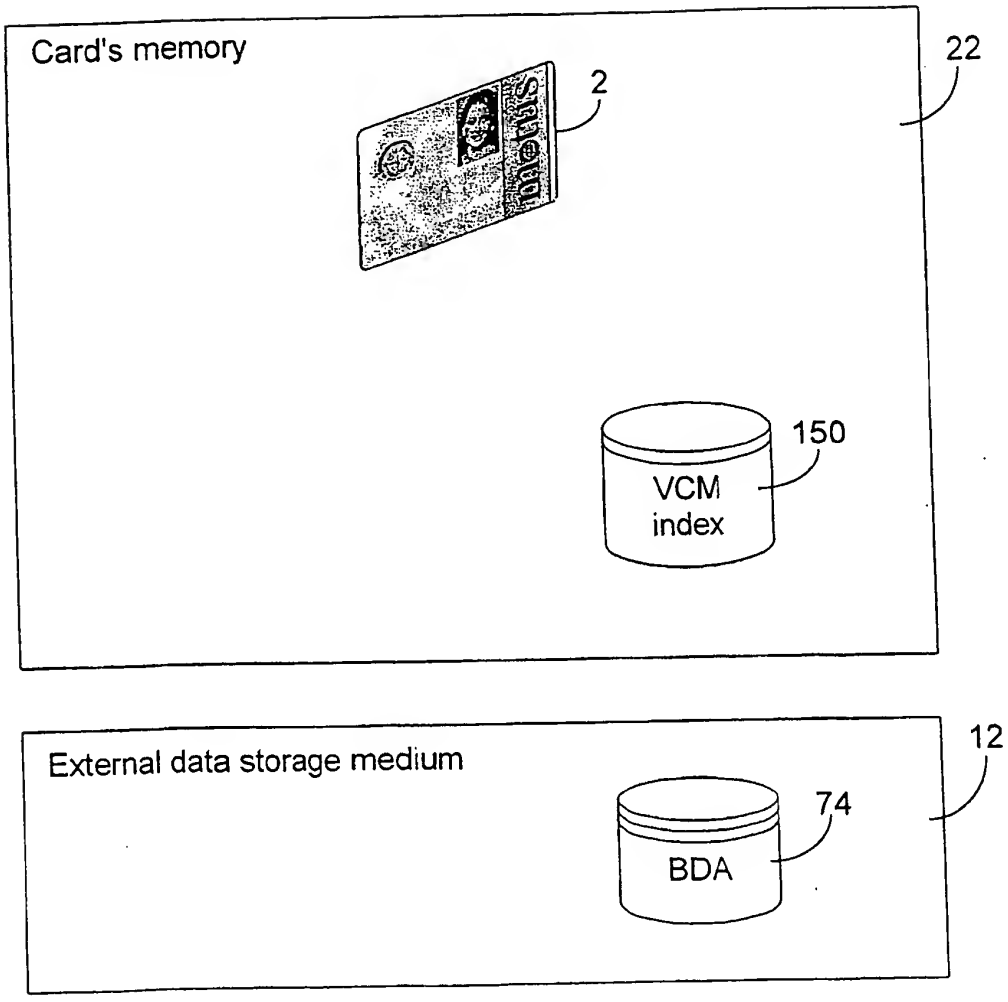

Card's memory
22
2
150
VCM
index
External data storage medium
12
74
BDA

Fig. 12

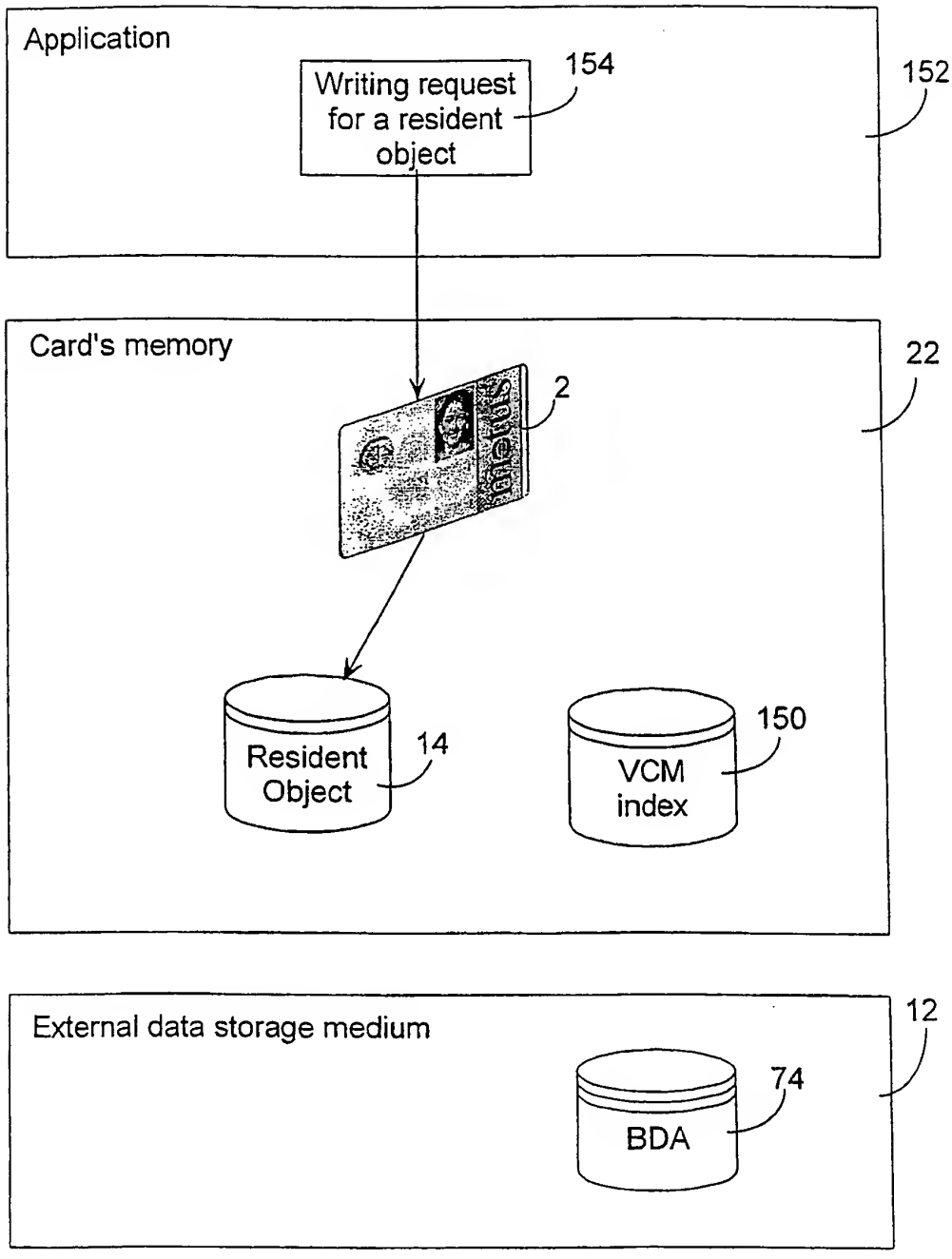

Application
Writing request for a resident object
154
152
Card's memory
22
2
Resident Object
14
VCM index
150
External data storage medium
12
BDA
74

Fig. 13

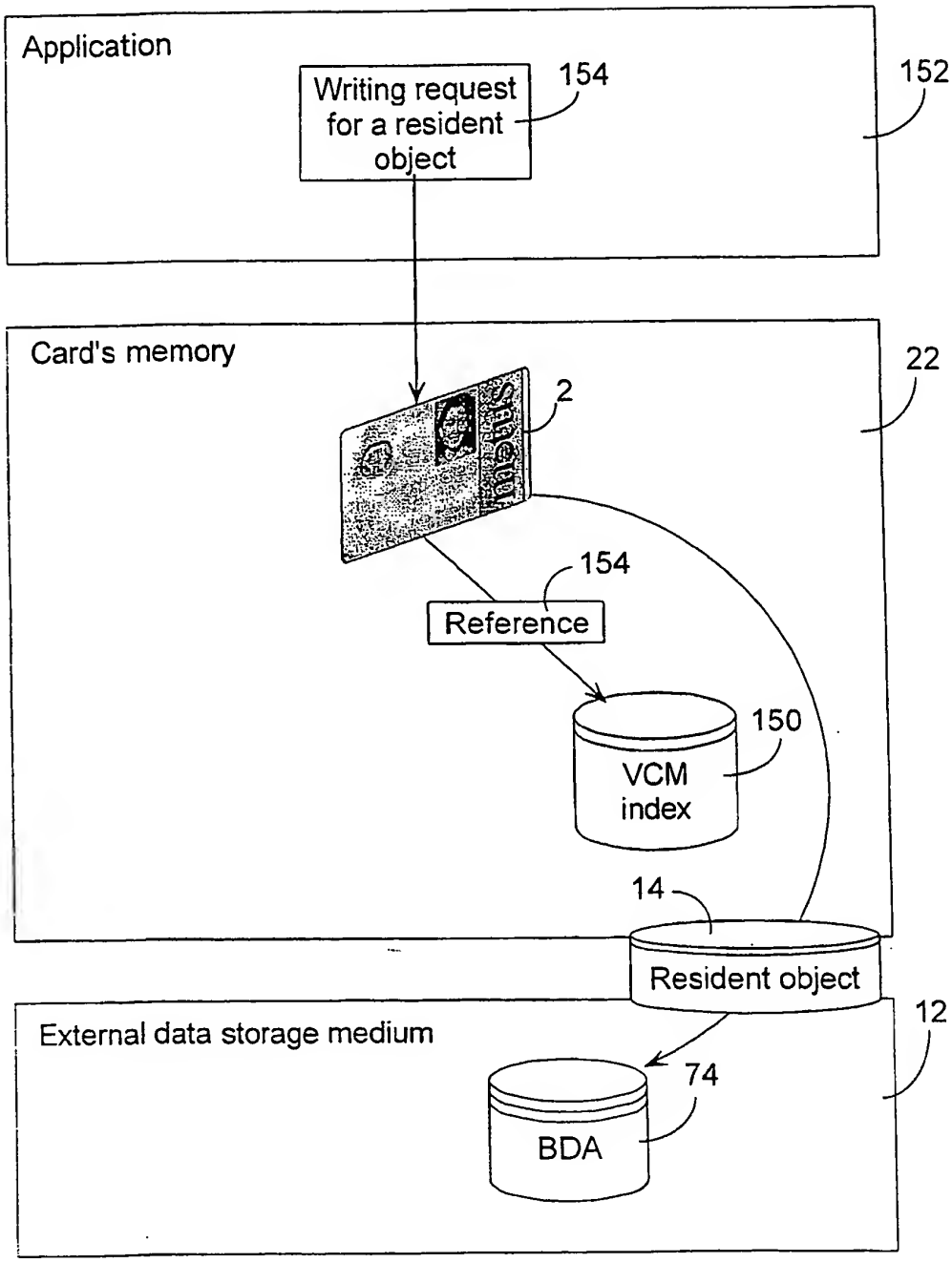

Application
Writing request for a resident object
154
152
Card's memory
22
2
154
Reference
150
VCM index
14
Resident object
External data storage medium
12
74
BDA

Fig. 14

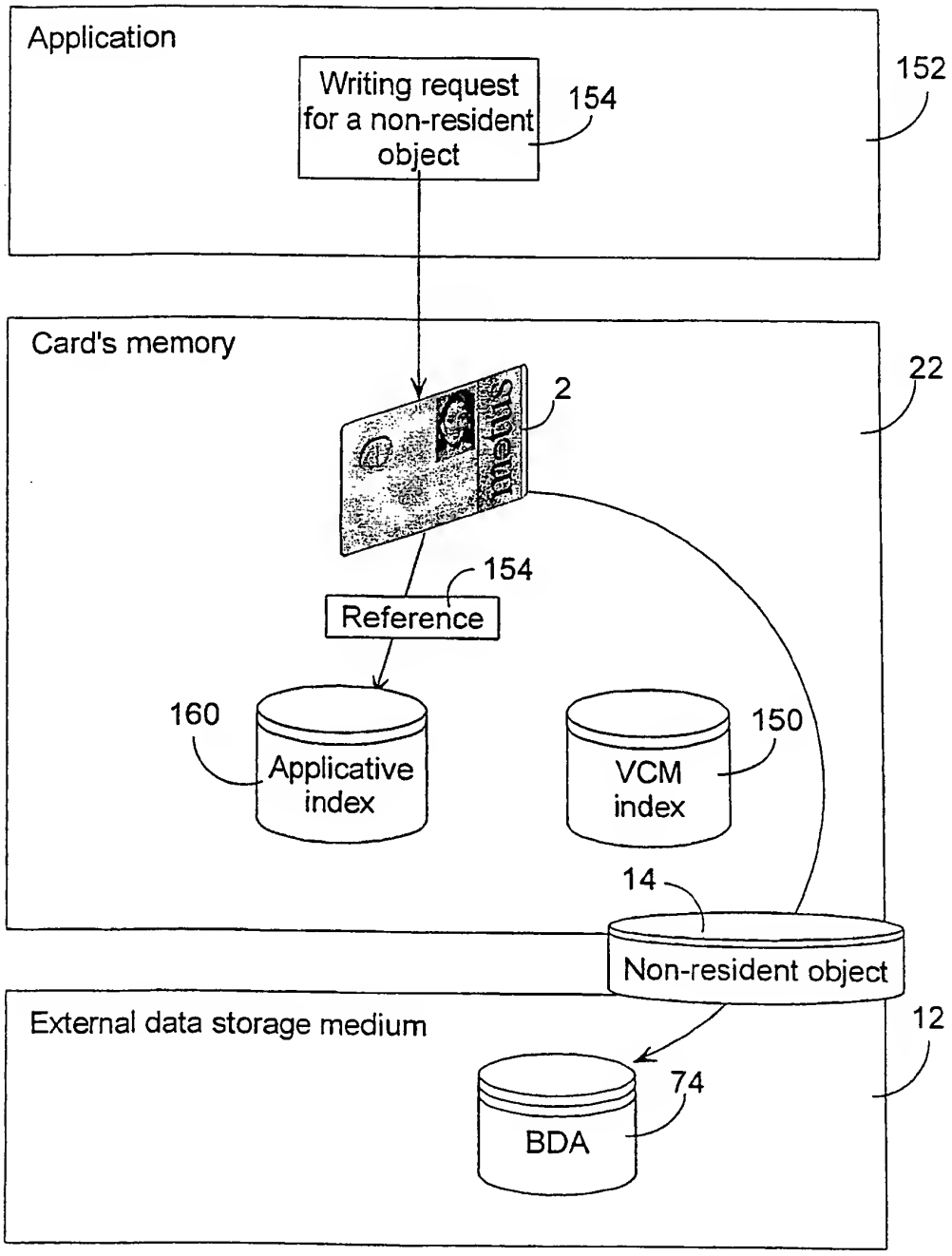

Application
Writing request for a non-resident object
154
152
Card's memory
22
2
154
Reference
160
Applicative index
150
VCM index
14
Non-resident object
External data storage medium
12
74
BDA

Fig. 15

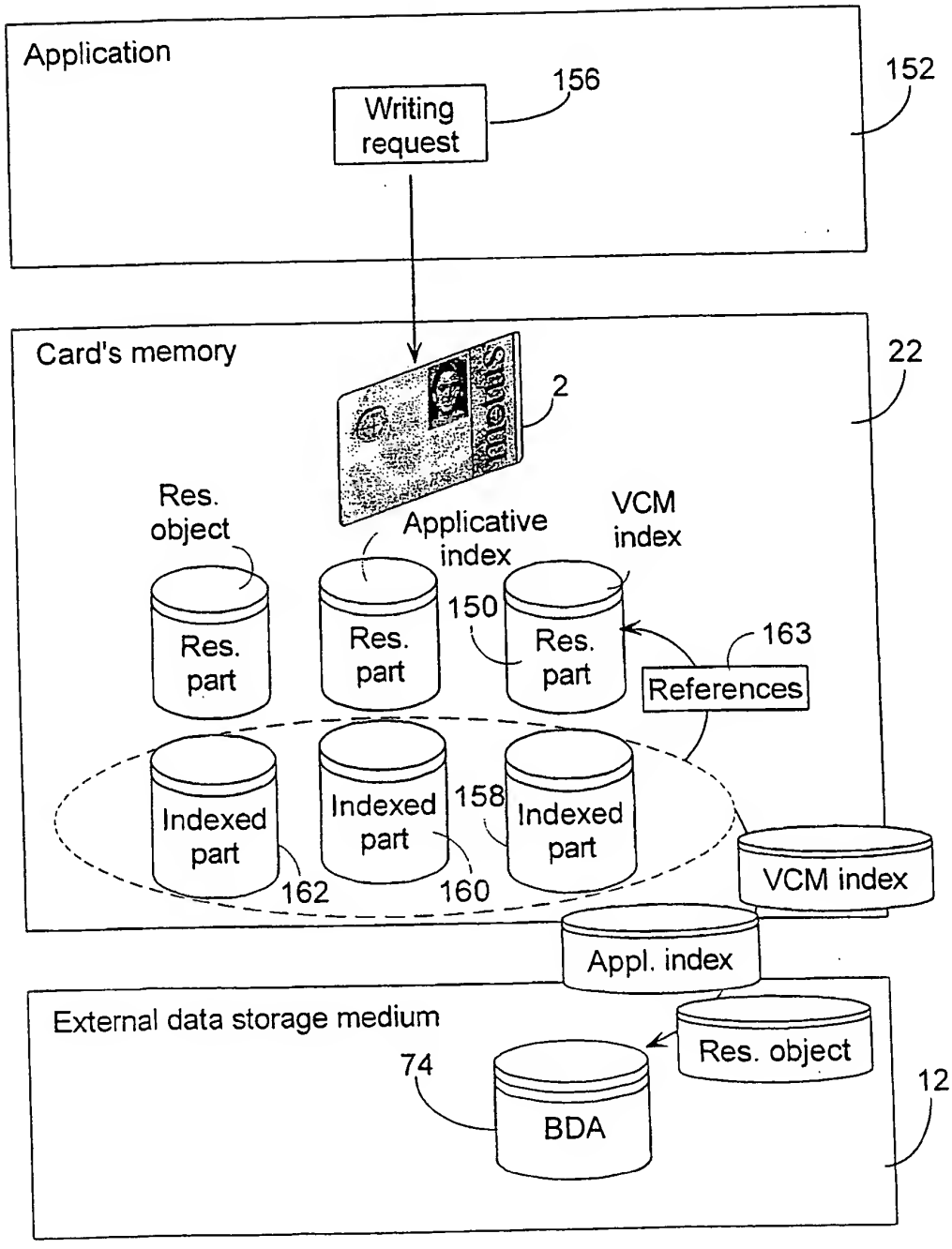

Application
Writing request
156
152
Card's memory
22
2
Res. object
Applicative index
VCM index
Res. part
Res. part
Res. part
150
163
References
Indexed part
Indexed part
Indexed part
158
162
160
VCM index
Appl. index
External data storage medium
Res. object
74
BDA
12

Fig. 16

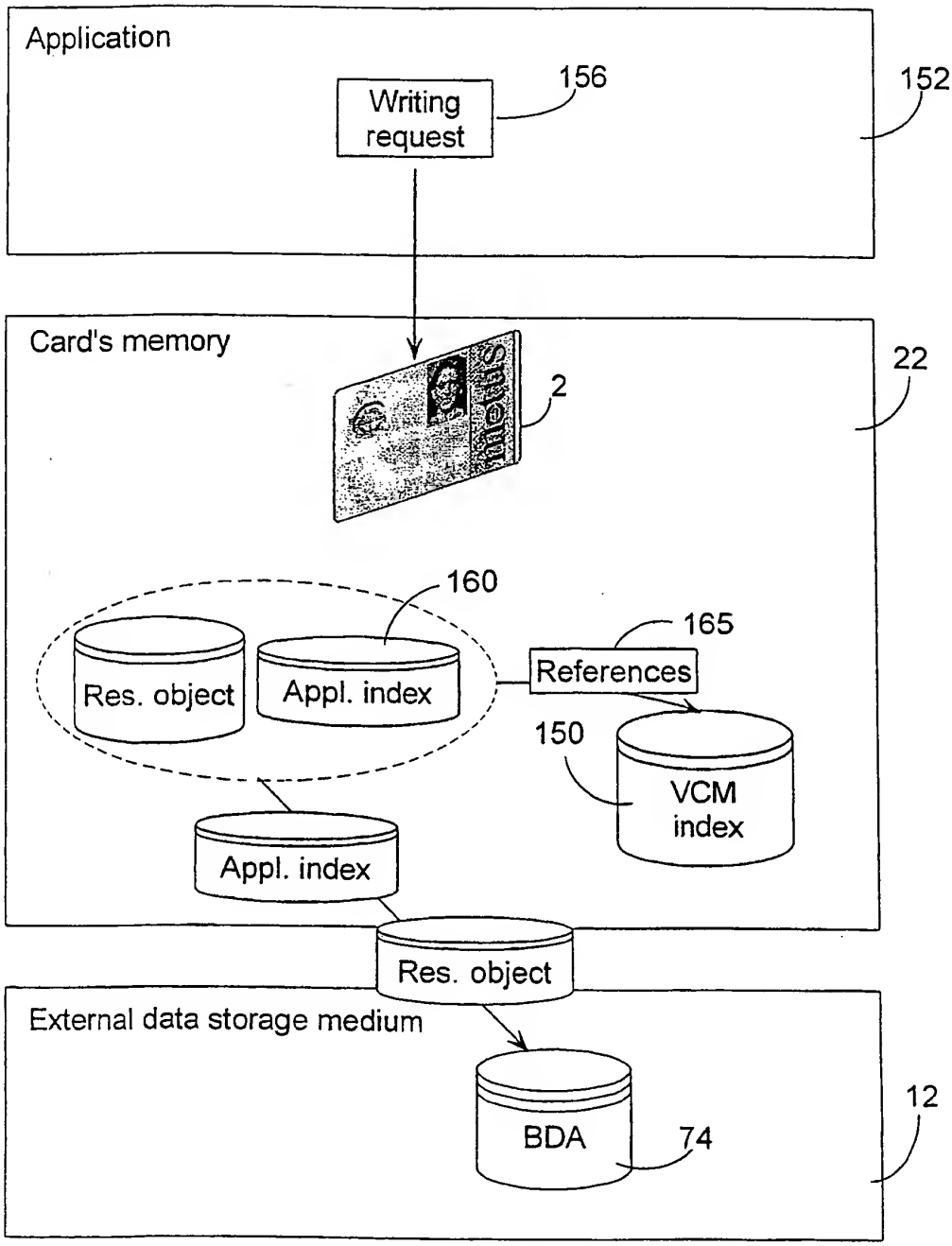

Application
Writing request
156
152
Card's memory
22
2
160
165
Res. object
Appl. index
References
150
VCM index
Appl. index
Res. object
External data storage medium
12
BDA
74

Fig. 17

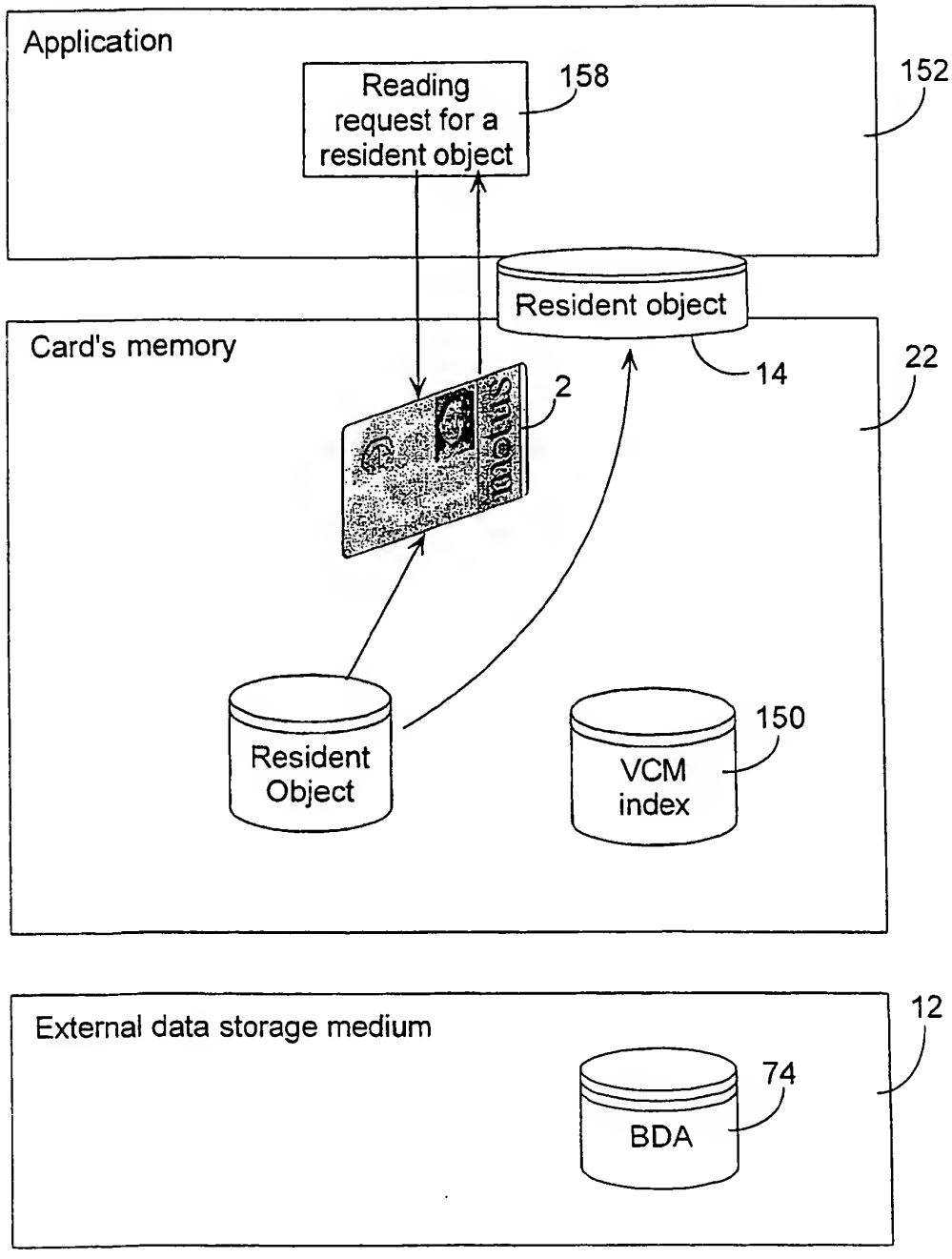

Application
Reading request for a resident object
158
152
Resident object
Card's memory
14
22
2
Resident Object
VCM index
150
External data storage medium
12
74
BDA

Fig. 18

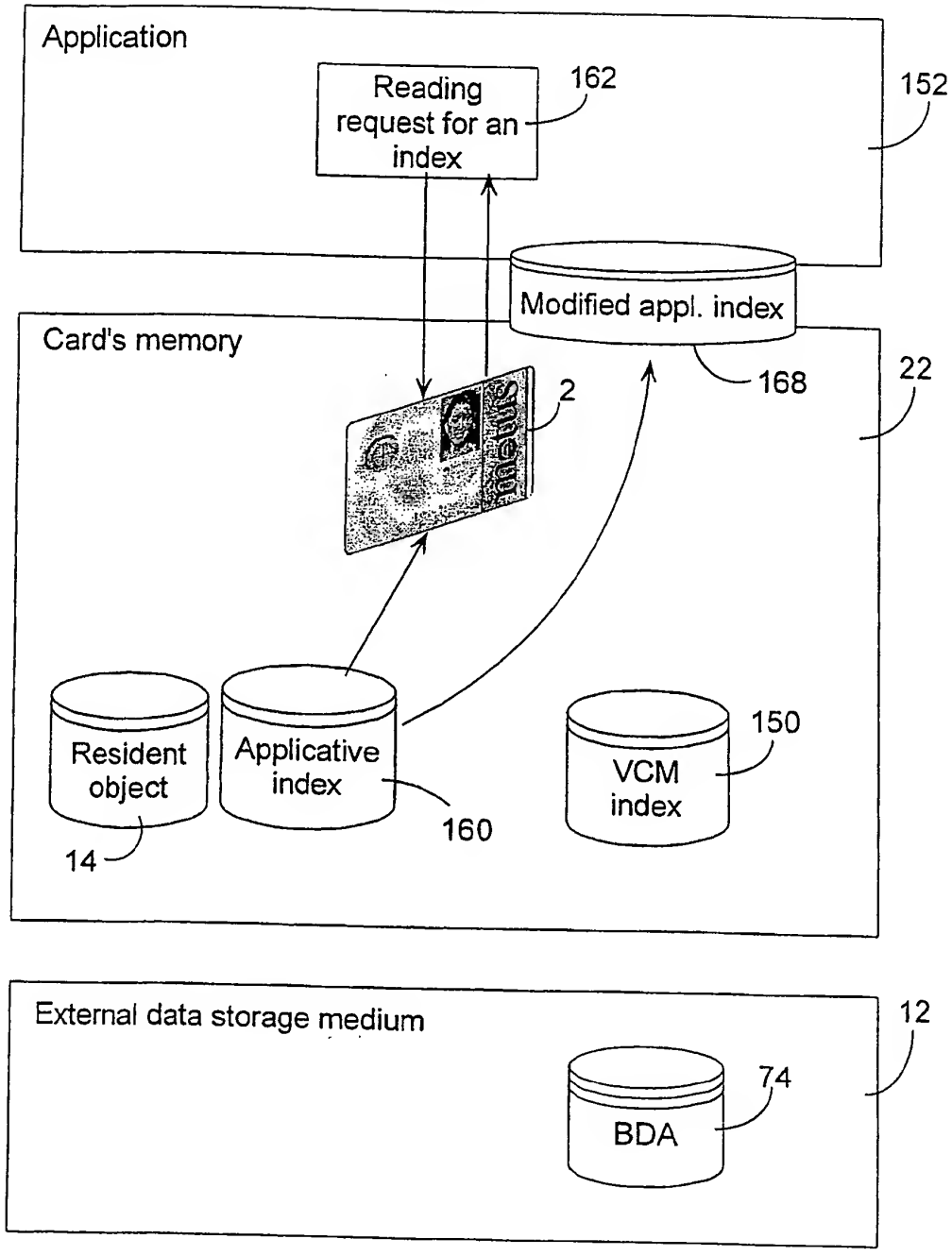

Application
Reading request for an index
162
152
Modified appl. index
Card's memory
168
22
2
Resident object
14
Applicative index
160
VCM index
150
External data storage medium
12
BDA
74

Fig. 19

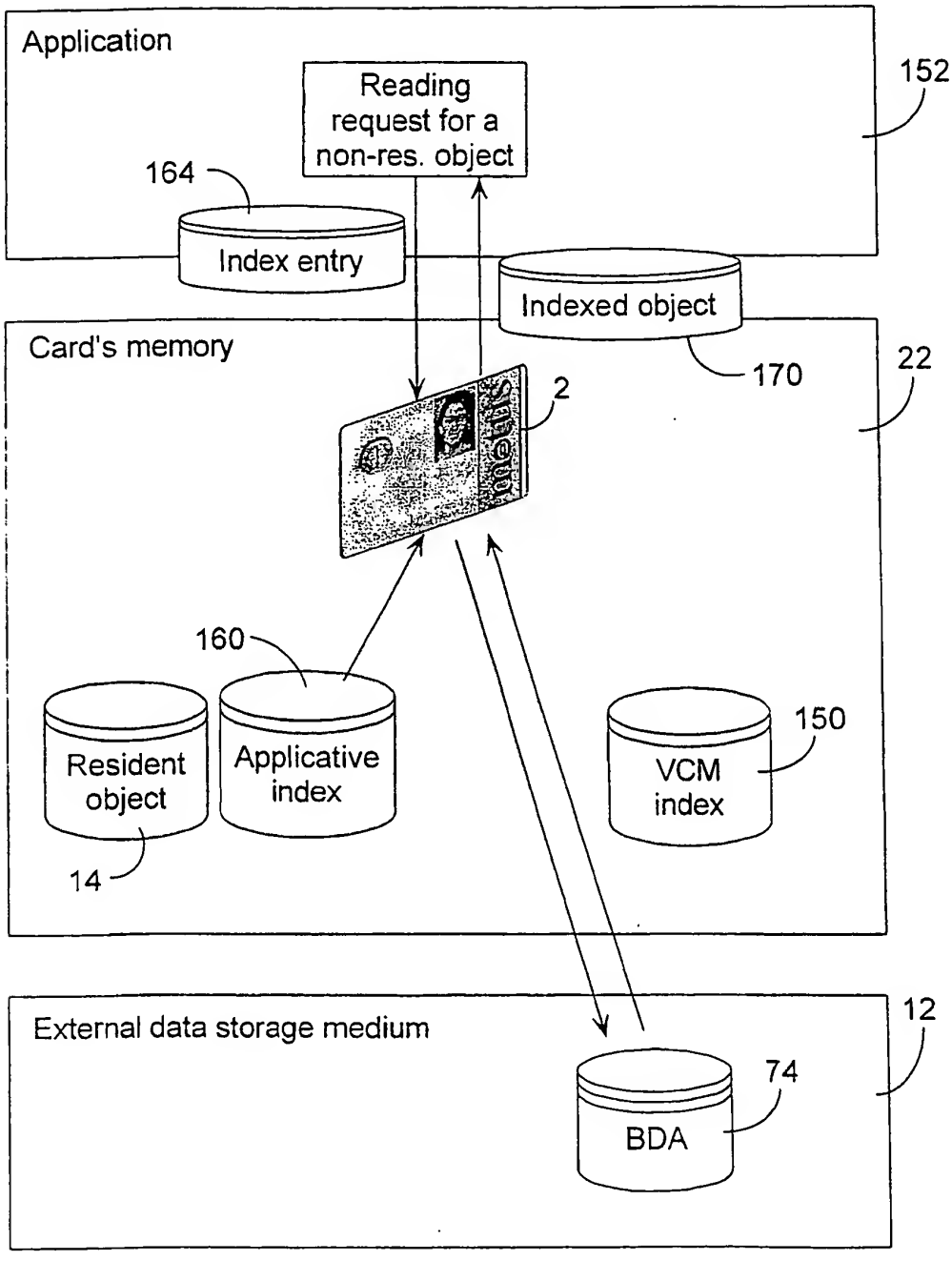

Application
Reading request for a non-res. object
152
164
Index entry
Indexed object
Card's memory
22
170
2
160
Resident object
Applicative index
VCM index
150
14
External data storage medium
12
74
BDA

Fig. 20

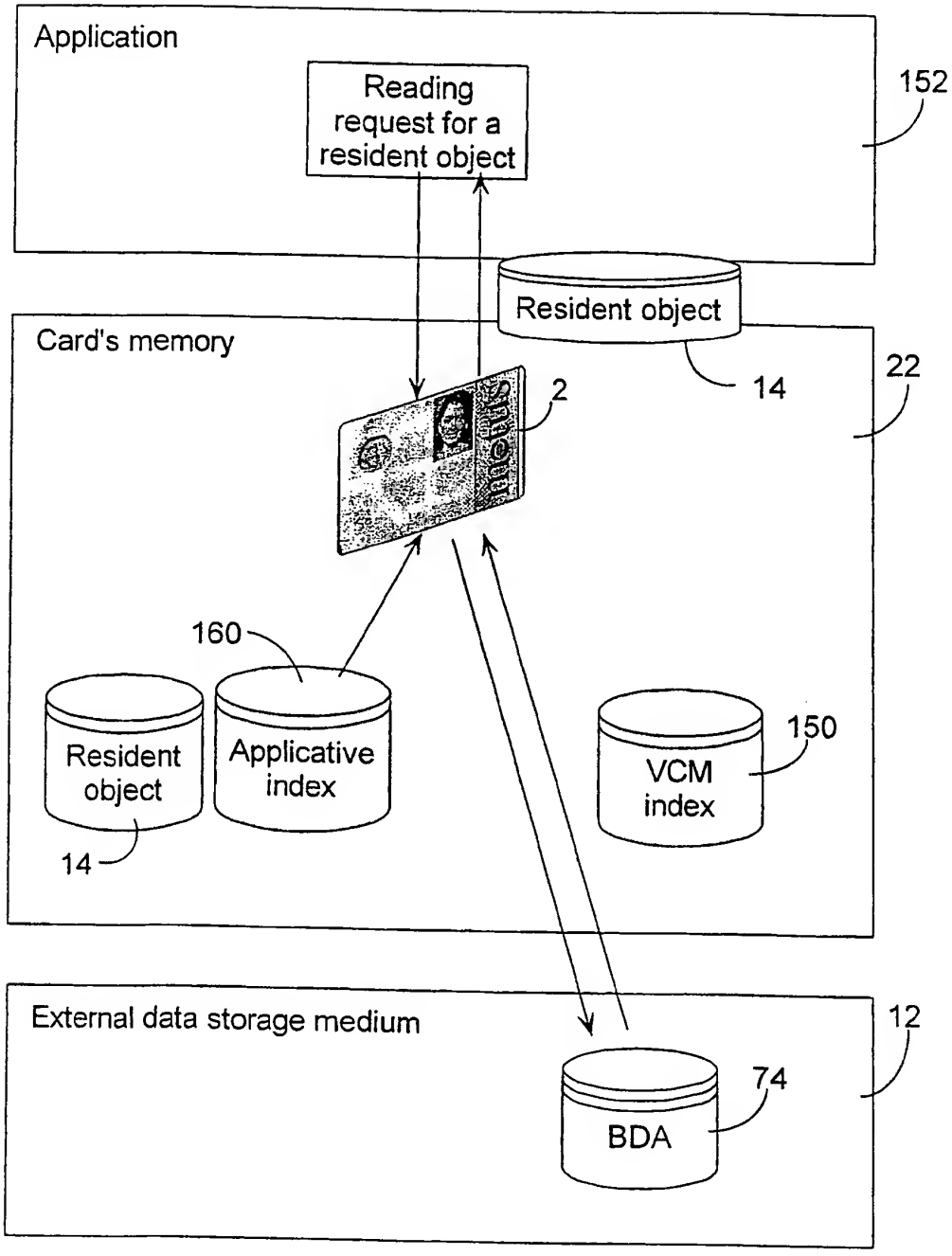

Application
Reading request for a resident object
152
Resident object
14
Card's memory
22
2
160
Resident object
14
Applicative index
VCM index
150
External data storage medium
12
BDA
74

Fig. 21

INTERNATIONAL SEARCH REPORT

Inte onal Application No
PCT/CA 99/01011

A. CLASSIFICATION OF SUBJECT MATTER
IPC 7 G07F7/10 G06F1/00 G06F12/14

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
IPC 7 G07F G06F

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	EP 0 856 818 A (MOTOROLA INC) 5 August 1998 (1998-08-05) the whole document	1-3, 9-23, 25-27
Y	column 2, line 39 -column 4, line 17	4-8,24, 28-40
X	EP 0 851 359 A (SGS THOMSON MICROELECTRONICS) 1 July 1998 (1998-07-01) column 4, line 1 -column 5, line 40; figure 1	1
Y	US 4 960 982 A (TAKAHIRA KENICHI) 2 October 1990 (1990-10-02)	1
Y	column 4, line 7 -column 5, line 19; figure 2	1
Y	column 5, line 13-19	24

-/--

[X] Further documents are listed in the continuation of box C.

[X] Patent family members are listed in annex.

* Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier document but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
18 February 2000	28/02/2000

Name and mailing address of the ISA	Authorized officer
European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Tx. 31 651 epo nl, Fax (+31-70) 340-3016	Weber, R

1

Form PCT/ISA/210 (second sheet) (July 1992)

C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	EP 0 330 404 A (FUJITSU LTD) 30 August 1989 (1989-08-30)	1
Y	column 4, line 11-33; figure 4	1
Y	column 4, line 21-33	28-40
Y	WO 97 29416 A (INTEGRATED TECH AMERICA ;BRADLEY JAMES V (US); MOONEY DAVID M (US)) 14 August 1997 (1997-08-14)	1
Y	page 5, line 14-19	1
Y	page 17, line 12 -page 18, line 14	4-8
Y	page 12, paragraph 2	24
Y	page 15, line 4-9	28-40
Y	US 5 784 459 A (ZUKOWSKI DEBORRA JEAN ET AL) 21 July 1998 (1998-07-21)	1
Y	column 4, line 10-30	28-40

1

INTERNATIONAL SEARCH REPORT

information on patent family members

Int ional Application No
PCT/CA 99/01011

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
EP 0856818 A	05-08-1998	GB 2321728 A	05-08-1998
		CN 1195827 A	14-10-1998
		JP 10222425 A	21-08-1998
EP 0851359 A	01-07-1998	FR 2757654 A	26-06-1998
		DE 69700263 D	15-07-1999
		DE 69700263 T	20-01-2000
		US 6002619 A	14-12-1999
US 4960982 A	02-10-1990	JP 63253493 A	20-10-1988
		DE 3811378 A	27-10-1988
		FR 2613856 A	14-10-1988
EP 0330404 A	30-08-1989	JP 1213711 A	28-08-1989
		JP 2534532 B	18-09-1996
		JP 2005158 A	10-01-1990
		DE 68919483 D	12-01-1995
		ES 2064432 T	01-02-1995
		US 4985920 A	15-01-1991
WO 9729416 A	14-08-1997	AU 2119697 A	28-08-1997
		CA 2245822 A	14-08-1997
		EP 0885417 A	23-12-1998
US 5784459 A	21-07-1998	NONE	